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 47τ KB for τ 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 90 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 [9]. Example use-cases are electronic voting [1], mixing networks [14], DC-networks [10] and fully homomorphic encryption [15]. 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 protocols achieving security even against quantum adversaries, see, e.g., the shuffles by Aranha *et al.* [4] and Costa *et al.* [11]. However, there are few constructions that provide verifiable decryption for these 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 [8]. 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 moduli. 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. To show its practicality, we give concrete parameters and estimate the size in Section 4.1 and give timings from our proof-of-concept implementation in Section. 4.2.

1.2 Related Work

We compare to the verifiable decryption schemes for lattices by Lyubashevsky *et al.* [17], Gjøsteen *et al.* [13] and Boschini *et al.* [7] in Section. 4.3.

2 Lattice-Based Cryptography

Let N be a power of 2 and $q = 1 \mod 2N$ 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 $\left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$, and compute inner products $\langle \cdot, \cdot \rangle$ and norms as vectors over $\mathbb{Z}$:

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

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

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

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

Lyubashevsky *et al.* [17] suggest to require that $\langle \boldsymbol{z}, \boldsymbol{v}\rangle \geq 0$. Then we can set $M = \exp(\|v\|_2/2\sigma^2)$. For $\sigma = 0.675\|\boldsymbol{v}\|_2$, we get $M \approx 3$, with the effect of rejecting about half of the vectors up front. See [17, Figure 2] for details.

2.2 Hardness Assumptions

We first define the Search Knapsack problem in the ℓ_2 norm, also denoted as SKS^2. The SKS^2 problem is the Ring-SIS problem in its Hermite Normal Form.

Definition 1. *The $\mathsf{SKS}^2_{N,q,\beta}$ problem is to find a short vector $\boldsymbol{x}$ of ℓ_2 norm less than or equal to β in R_q^2 satisfying $[\, a \quad 1 \,] \cdot \boldsymbol{x} = 0$ for a given uniformly random a in R_q. An algorithm $\mathcal{A}$ has advantage ϵ in solving the $\mathsf{SKS}^2_{N,q,\beta}$ problem if*

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

We also define the Decisional Knapsack problem (DKS^∞) in the ℓ_∞ norm. DKS^∞ is equivalent to the Ring-LWE problem when the number of samples is limited.

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

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

See [16, 18, 19] 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 $\mathtt{D}$ be a bounded distribution over R_q, let $\beta_\infty \in \mathbb{N}$ be a bound and let λ be the security parameter. The BGV encryption scheme [8] consists of three algorithms: key generation (KGen), encryption (Enc) and decryption (Dec), where

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

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

2.4 Lattice-Based Commitments

Let $\mathcal{N}_{\sigma_\mathrm{C}}$ be a Gaussian distribution over R_q with standard deviation σ_C. The commitment scheme by Baum *et al.* [6] consists of three algorithms: key generation (KGen), committing (Com) and opening (Open), where

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

$$\begin{aligned} \boldsymbol{A}_1 &= \begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{A}_1' \end{bmatrix} & \text{where } \boldsymbol{A}_1' &\leftarrow\$\, R_q^{n\times(k-n)} \\ \boldsymbol{a}_2 &= \begin{bmatrix} \boldsymbol{0}^n & 1 & \boldsymbol{a}_2' \end{bmatrix} & \text{where } \boldsymbol{a}_2' &\leftarrow\$\, R_q^{(k-n-1)}, \end{aligned}$$

  for height $n+1$ and width k and let $\mathtt{pk}$ be $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{a}_2 \end{bmatrix}$.
- Com commits to messages $m \in R_q$ by sampling $\boldsymbol{r}_m \leftarrow\$\, S_{\beta_\infty}$, and computes

$$\mathtt{Com}_{\mathtt{pk}}(m; \boldsymbol{r}_m) = \boldsymbol{A} \cdot \boldsymbol{r}_m + \begin{bmatrix} \boldsymbol{0} \\ m \end{bmatrix} = \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{bmatrix} = [\![m]\!].$$

  Com outputs commitment $[\![m]\!]$ and opening $\boldsymbol{d} = (m, \boldsymbol{r}_m, 1)$.
- Open verifies whether $(m, \boldsymbol{r}_m, f)$, with $f \in \bar{\mathcal{C}}$, is a valid opening of $[\![m]\!]$ with respect to $\mathtt{pk}$ by checking that $\|\boldsymbol{r}_m[i]\|_2 \leq 4\sigma_{\mathrm{C}}\sqrt{N}$, for $i \in [k]$, and if

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

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

The commitment scheme is hiding if the $\mathsf{DKS}^\infty_{N,q,\beta_\infty}$ problem is hard and it is binding if the $\mathsf{SKS}^2_{N,q,16\sigma_{\mathrm{C}}\sqrt{\nu N}}$ problem is hard, see [6, Section 4].

2.5 Zero-Knowledge Proof of Linear Relations

Let $[\![y]\!], [\![y']\!]$ be commitments as above such that $y' = \alpha y + \beta$ for some public values $\alpha, \beta \in R_q$. The protocol $\mathit{\Pi}_{\mathrm{LIN}}$ in [4, Figure 1] is a zero-knowledge proof of knowledge, with ℓ_2 bound $B_{\mathrm{C}} = 2\sigma_{\mathrm{C}}\sqrt{N}$ on the responses $\boldsymbol{z}_i$, for the relation:

$$\mathcal{R}_{\mathrm{Lin}} = \left\{ (x, w) \;\middle|\; \begin{array}{l} x = (\alpha, \beta, [\![y]\!], [\![y']\!]), w = (y, \boldsymbol{r}_y, \boldsymbol{r}_{y'}, f, f') : \\ \mathtt{Open}([\![y]\!], y, \boldsymbol{r}_y, f) = \mathtt{Open}([\![y']\!], \alpha \cdot y + \beta, \boldsymbol{r}_{y'}, f') = 1 \end{array} \right\}$$

When applying the Fiat-Shamir transform [12], 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 $\pi_L = (c, \boldsymbol{z}_1, \boldsymbol{z}_2)$, where each $\boldsymbol{z}_i$ is of size $kN\log_2(6\sigma_{\mathrm{C}})$ bits. We can compress each $\boldsymbol{z}_i$ to get a proof of total size $2(k-n)N\log_2(6\sigma_{\mathrm{C}})$ bits by checking an approximate equality instead, as described in [4, Section 3.2]. We denote by

$$\begin{aligned} \pi_L &\leftarrow \mathit{\Pi}_{\mathrm{LIN}}((y, \boldsymbol{r}_y, \boldsymbol{r}_{y'}, f_y, f_{y'}); (\alpha, \beta, [\![y]\!], [\![y']\!])), \text{ and} \\ 0 \vee 1 &\leftarrow \mathit{\Pi}_{\mathrm{LINV}}((\alpha, \beta, [\![y]\!], [\![y']\!]); \pi_L), \end{aligned}$$

the run of the proof and verification protocols, respectively, where the verification protocol $\mathit{\Pi}_{\mathrm{LinV}}$ performs the checks as in the last step in [4, Figure 1] and also verifies that c was computed correctly with respect to the transcript. $\mathit{\Pi}_{\mathrm{Lin}}$ is a sound proof of knowledge in the ROM if the $\mathsf{SKS}^2_{N,q,2B_{\mathrm{C}}}$ problem is hard.

2.6 Amortized Zero-Knowledge Proof of Bounded Openings

Let $\boldsymbol{A}$ be a publicly known $r \times v$-matrix over R_q, let $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_\tau$ be bounded elements in R_q^v and let $\boldsymbol{A}\boldsymbol{s}_i = \boldsymbol{t}_i$ for $i \in [\tau]$. Letting $\boldsymbol{S}$ be the matrix whose columns are $\boldsymbol{s}_i$ and $\boldsymbol{T}$ be the equivalent matrix for $\boldsymbol{t}_i$, Baum *et al.* [5] give a efficient amortized zero-knowledge proof of knowledge for the relation:

$$\mathcal{R}_{\text{A}} = \left\{ (x, w) \;\middle|\; \begin{array}{c} x = (\boldsymbol{A}, \boldsymbol{T}), w = \boldsymbol{S} : \\ \forall i \in [\tau] : \; \boldsymbol{t}_i = \boldsymbol{A}\boldsymbol{s}_i \wedge ||\boldsymbol{s}_i||_2 \leq 2 \cdot B_{\text{A}} \end{array} \right\}$$

The protocol $\mathit{\Pi}_{\text{A}}$ is depicted in [5, Figure 1]. We use a challenge matrix $\boldsymbol{C}$ with entries sampled from the set $\mathcal{C}_{\text{A}} = \{0, 1\}$. For security parameter λ, we define the number of parallel protocol instances to be $\hat{n} = \lambda + 2$. Denote by

$$\pi_{\text{A}} \leftarrow \mathit{\Pi}_{\text{A}}(\boldsymbol{S}; (\boldsymbol{A}, \boldsymbol{T})), \text{ and } 0 \vee 1 \leftarrow \mathit{\Pi}_{\text{AV}}((\boldsymbol{A}, \boldsymbol{T}); \pi_{\text{A}}),$$

the run of the proof and verification protocols, respectively, where the $\mathit{\Pi}_{\text{A}}$-protocol, using Fiat-Shamir, produces a proof of the form $\pi_{\text{A}} = (\boldsymbol{C}, \boldsymbol{Z})$, where $\boldsymbol{C}$ is the output of a hash-function applied to the full transcript, and the $\mathit{\Pi}_{\text{AV}}$-protocol consists of the two checks in the last step in [5, Figure 1]. The verification bound on each column of $\boldsymbol{Z}$ is $B_{\text{A}} = \sqrt{2vN}\sigma_{\text{A}}$. Note that σ_{A}, and also B_{A}, depends on the norm of $\boldsymbol{S}$ (see rejection sampling Section 2.1). Hence, the bound we can prove depends on the number of equations in the statement. $\mathit{\Pi}_{\text{A}}$ is a sound proof of knowledge in the ROM if the $\mathsf{SKS}^2_{N,q,2B_{\text{A}}}$ problem 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 $m = (v - us \mod q) \mod p$. Then, he commits to the noise $d = er + e'' - se'$ in the ciphertexts as $[\![d]\!]$. Finally, he proves two statements in zero-knowledge: 1) the linear relation $p[\![d]\!] = v - m - u[\![s]\!]$ holds modulo q with respect to the noise and a public commitment to the secret key, and 2) the value committed to in $[\![d]\!]$ is shorter than some bound $B < q/2p$.

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

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

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 $[\![s]\!]$, using any exact proof from the literature.

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

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

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

A verifier $\mathcal{V}$ runs the verification protocol Π_{DecV} which checks that all proofs $\{\pi_{L_i}\}_{i=1}^{\tau}$ and π_{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 Π_{Dec} is a complete, sound and zero-knowledge proof protocol in the ROM for relation R_{Dec} when $B_{\text{A}} < q/(4p\sqrt{N})$.*

Proof. We prove each of the properties as following:

Completeness. It follows directly that Π_{Dec} is complete if the encryption scheme is correct, which is the case when $\|v - su\| < q/2$, and the protocols Π_{LIN} and Π_{A} are complete. Hence, we only need to make sure that $\|v - su\| < q/2$. The protocol Π_{A} guarantees that the noise is bounded as $\|d_i\|_2 \leq 2B_{\text{A}}$. It follows that if $B_{\text{A}} < q/(4p\sqrt{N})$ 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 Π_{LIN} and Π_{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 SKS^2 problem for the given parameters.

Honest-verifier zero-knowledge. The zero-knowledge property follows directly from the underlying zero-knowledge protocols Π_{LIN} and Π_{A}, which are both honest-verifier zero-knowledge. Hence, with input messages $m_1, \ldots, m_\tau$ we can simulate the decryption proof by sampling uniformly random values d_i, committing to them as $[\![d_i]\!]$ and then simulating all the proofs π_{L_i} and π_{A} subsequently.

4 Performance

4.1 Parameters and Size

From the verifiable decryption protocol in Section 3 we get that the statement consists of τ ciphertexts (u_i, v_i) and messages m_i. Each element u_i and u_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 π_L are of size $2(k-n)N \log_2(6\sigma_{\text{C}})$ bits, for $\sigma_{\text{C}} = 11\nu\beta_\infty\sqrt{kN}$, and the proof π_{A} is of size $(k+1)\hat{n}N \log_2(6\sigma_{\text{A}})$ bits. However, the norm bound B_{A} depends on the number of equations being proved at once, and hence, if τ 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 $\mathtt{D}$ be the ternary distribution over R_q. It then follows that for honestly generated ciphertexts

p	q	N	β_∞	M	k	n	$\hat{n}$	ν	σ_{C}	B_{C}	σ_{A}	B_{A}
2	$\approx 2^{50}$	2048	1	3	3	1	130	36	$\approx 2^{15.9}$	$\approx 2^{22.4}$	$\approx 2^{34.4}$	$\approx 2^{41.5}$

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

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

$$\begin{aligned}\|d_i\|_\infty &\leq 2\sqrt{N}B_{\mathrm{A}} = \sqrt{8(k+1)N\tau}\sigma_{\mathrm{A}} \leq \sqrt{8(k+1)N\tau}\cdot 0.675\left\|\boldsymbol{S}'\boldsymbol{C}'\right\|_2 \\ &\leq 2\sqrt{(k+1)N\tau}\cdot(4kN\sqrt{N}\sigma_{\mathrm{C}} + p(2N+1)) \\ &\leq 2\sqrt{(k+1)N\tau}\cdot(4kN\sqrt{N}\cdot 11\cdot\nu\cdot\sqrt{kN} + p(2N+1)).\end{aligned}$$

Thus, setting $k = 3, n = 1, \hat{n} = 130, \nu = 36$ and $\tau = N = 2048$ gives us $\|d_i\|_\infty \approx 2^{48}$, and we can safely set $q \approx 2^{50}$ to get correctness. We claim at least 128 bits security against a quantum adversary for these parameters using the LWE estimator by Albrecht *et al.* [3] 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 $[\![d_i]\!]$	Proof π_{L_i}	Proof π_{A}	Proof π_{DEC}
0.256 KB	25.6 KB	25.6 KB	19 KB	2.4τ KB	47τ KB

Table 2. Sizes for parameters $p = 2, q \approx 2^{50}$ and $N = 2048$ computing proof $\pi_{\mathrm{DEC}} = (\{[\![d_i]\!], \pi_{L_i}\}_{i=1}^{\tau}, \pi_{\mathrm{A}})$, where shortness proofs π_{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 [20]. 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 250 lines of code, and is available online[⋆]. A comparison of NTL to NFLlib [2] indicates that an optimized implementation could provide speedup by at least an order of magnitude.

Noise $[\![d_i]\!]$	Proof Π_{LIN}	Verification Π_{LINV}	Proof Π_{A}	Verification Π_{AV}	Proof π_{DEC}
6τ ms	59τ ms	15τ ms	25τ ms	12τ ms	90τ 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.* [17] and Gjøsteen *et al.* [13]. As noted by [13, Section 8], the protocol by Boschini *et al.* [7] give proof sizes of approximate 90 KB, which is roughly twice the size of π_{DEC}. Furthermore, the run time is several minutes per ciphertext, so this would deem it unusable for moderate or large sets of ciphertexts.

[⋆] 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 $B_\infty = 2$. The proof of correct decryption is of size 43.6 KB. We note that our proof is of approximately 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 50 bits. Furthermore, our protocol is much simpler, as [17] 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.* (EPRINT 2021). They give a verifiable decryption protocol $\mathit{\Pi}_{\texttt{ZKPCD}}$ for the BGV encryption scheme. However, because of their noise drowning techniques, they are forced to use a moduli of at least $q \approx 2^{62}$. Their proof size is also depending on the soundness parameter λ, giving a proof of size 16λ KB per ciphertext. For an interactive protocol with $\lambda = 10$ they get a proof of size $3.5\times$ larger than our proof, and for a non-interactive protocol with $\lambda = 100$ their proof size is $35\times$ larger than ours. They have not implemented their protocol, but estimate a cost of at least $\approx 100\lambda$ μs per ciphertext using NFLlib [2], which is similar to our protocol for reasonable instances of λ.

They also sketch a protocol $\mathit{\Pi}_{\texttt{DistDec}}$ [13, Section 8], requiring $q \approx 2^{110}$ and $N = 4096$. This protocol gives a proof of size ≈ 516 KB per ciphertext, a factor 11 larger than our proof. They do not provide timings for this protocol.

References

1. Adida, B.: Helios: Web-based open-audit voting. In: van Oorschot, P.C. (ed.) USENIX Security 2008. pp. 335–348. USENIX Association (Jul / Aug 2008)
2. Aguilar Melchor, C., Barrier, J., Guelton, S., Guinet, A., Killijian, M.O., Lepoint, T.: NFLlib: NTT-based fast lattice library. In: Sako, K. (ed.) CT-RSA 2016. LNCS, vol. 9610, pp. 341–356. Springer, Heidelberg (Feb / Mar 2016). https://doi.org/10.1007/978-3-319-29485-8_20
3. Albrecht, M.R., Player, R., Scott, S.: On the concrete hardness of learning with errors. Journal of Mathematical Cryptology **9**(3), 169–203 (2015)
4. Aranha, D.F., Baum, C., Gjøsteen, K., Silde, T., Tunge, T.: Lattice-based proof of shuffle and applications to electronic voting. In: Paterson, K.G. (ed.) CT-RSA 2021. LNCS, vol. 12704, pp. 227–251. Springer, Heidelberg (May 2021). https://doi.org/10.1007/978-3-030-75539-3_10
5. Baum, C., Bootle, J., Cerulli, A., del Pino, R., Groth, J., Lyubashevsky, V.: Sublinear lattice-based zero-knowledge arguments for arithmetic circuits. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 669–699. Springer, Heidelberg (Aug 2018). https://doi.org/10.1007/978-3-319-96881-0_23
6. Baum, C., Damgård, I., Lyubashevsky, V., Oechsner, S., Peikert, C.: More efficient commitments from structured lattice assumptions. In: Catalano, D., De Prisco, R.

(eds.) SCN 18. LNCS, vol. 11035, pp. 368–385. Springer, Heidelberg (Sep 2018). https://doi.org/10.1007/978-3-319-98113-0_20

7. Boschini, C., Camenisch, J., Ovsiankin, M., Spooner, N.: Efficient post-quantum SNARKs for RSIS and RLWE and their applications to privacy. In: Ding, J., Tillich, J.P. (eds.) Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020. pp. 247–267. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-030-44223-1_14
8. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. In: Goldwasser, S. (ed.) ITCS 2012. pp. 309–325. ACM (Jan 2012). https://doi.org/10.1145/2090236.2090262
9. 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 (Aug 2003). https://doi.org/10.1007/978-3-540-45146-4_8
10. Corrigan-Gibbs, H., Wolinsky, D.I., Ford, B.: Proactively accountable anonymous messaging in verdict. In: King, S.T. (ed.) USENIX Security 2013. pp. 147–162. USENIX Association (Aug 2013)
11. Costa, N., Martínez, R., Morillo, P.: Lattice-based proof of a shuffle. In: Bracciali, A., Clark, J., Pintore, F., Rønne, P.B., Sala, M. (eds.) FC 2019 Workshops. LNCS, vol. 11599, pp. 330–346. Springer, Heidelberg (Feb 2019). https://doi.org/10.1007/978-3-030-43725-1_23
12. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO'86. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (Aug 1987). https://doi.org/10.1007/3-540-47721-7_12
13. Gjøsteen, K., Haines, T., Müller, J., Rønne, P., Silde, T.: Verifiable decryption in the head. Cryptology ePrint Archive, Report 2021/558 (2021), https://eprint.iacr.org/eprint-bin/getfile.pl?entry=2021/558&version=20210503:201150&file=558.pdf
14. Haines, T., Müller, J.: SoK: Techniques for verifiable mix nets. In: Jia, L., Küsters, R. (eds.) CSF 2020 Computer Security Foundations Symposium. pp. 49–64. IEEE Computer Society Press (2020). https://doi.org/10.1109/CSF49147.2020.00012
15. Luo, F., Wang, K.: Verifiable decryption for fully homomorphic encryption. In: Chen, L., Manulis, M., Schneider, S. (eds.) ISC 2018. LNCS, vol. 11060, pp. 347–365. Springer, Heidelberg (Sep 2018). https://doi.org/10.1007/978-3-319-99136-8_19
16. Lyubashevsky, V., Micciancio, D.: Generalized compact Knapsacks are collision resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006, Part II. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (Jul 2006). https://doi.org/10.1007/11787006_13
17. Lyubashevsky, V., Nguyen, N.K., Seiler, G.: Shorter lattice-based zero-knowledge proofs via one-time commitments. In: Garay, J. (ed.) PKC 2021, Part I. LNCS, vol. 12710, pp. 215–241. Springer, Heidelberg (May 2021). https://doi.org/10.1007/978-3-030-75245-3_9
18. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (May / Jun 2010). https://doi.org/10.1007/978-3-642-13190-5_1
19. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–166. Springer, Heidelberg (Mar 2006). https://doi.org/10.1007/11681878_8
20. Shoup, V.: Ntl: A library for doing number theory (2021), https://libntl.org/index.html