Efficient Commitments and Zero-Knowledge Protocols from Ring-SIS with Applications to Lattice-based Threshold Cryptosystems

Carsten Baum[1], Ivan Damgård[2], Sabine Oechsner[2], and Chris Peikert[3]

[1] Department of Computer Science, Bar-Ilan University
carsten.baum@biu.ac.il
[2] Department of Computer Science, Aarhus University
{ivan, oechsner}@cs.au.dk
[3] Department of Computer Science and Engineering, University of Michigan
cpeikert@umich.edu

Abstract. We present an additively homomorphic commitment scheme with hardness based on the Ring-SIS problem. Our construction is statistically hiding as well as computationally binding and allows to commit to a vector of ring elements at once. We define the ring SIS problem in the canonical embedding (rather than in the standard polynomial representation) and this allows us to get a stronger connection between breaking our binding property and known worst case results in ideal lattices.
We show how to instantiate efficient zero-knowledge protocols that can be used to prove a number of relations among these commitments, and apply these in the context of lattice-based threshold cryptosystems: we give a generic transformation that can be used with certain (Ring-)LWE-based encryption schemes to make their algorithms actively secure. We show how this transformation can be used to implement distributed decryption with malicious security as well as maliciously secure threshold key generation in an efficient way.

1 Introduction

Over the past several years, lattice-based cryptography has developed and matured rapidly. As this development continues, it is desirable to have a full suite of efficient lattice-based tools and protocols. This is particularly important since lattice problems are currently some of the promising "post-quantum" replacements for the discrete logarithm and factoring problems. Therefore, we want to construct standard cryptographic primitives such as encryption and commitment schemes, plus companion protocols such as zero-knowledge proofs or threshold key generation, in the lattice setting.

Commitment schemes [Blu82] are a key tool in the design of cryptographic protocols and have countless applications. In particular, when combined with zero-knowledge proofs, they can enforce "good" behavior by adversarial parties and make the design of protocols secure against malicious attacks easier. A prime example where these can be used is in the context of lattice-based

threshold encryption schemes [DF89], where multiple parties collaborate to generate a public/private key pair or decrypt ciphertexts using an interactive protocol. Such encryption schemes are applied, e.g., in the SPDZ MPC protocol [DPSZ12,DKL+13], where Damgård et al. used a variant of the cryptosystems from [LPR10,BV11,BGV12] in their preprocessing protocol. However, in [DPSZ12], the question of key generation was avoided by assuming that a common public key and shares of the corresponding secret key were already in place (due to the lack of an efficient, actively secure protocol). Later, [DKL+13] gave a key generation procedure, but it was only covertly secure. As for distributed decryption, both papers used a simple semi-honestly secure solution: decryption might produce an incorrect output when players cheat, and the surrounding protocol has to check this output and catch any errors later. This error checking leads to additional overhead in both computation and communication, which is unsatisfying.

This state of affairs clearly leaves several problems open: first, we would like to have maliciously secure distributed key generation protocols that are efficient and avoid the use of generic zero-knowledge techniques. Second, we would like to have distributed key generation protocols that do not have to rely on external checks for malicious security. In this paper we are interested in constructing a commitment scheme and efficient zero-knowledge protocols for proving relations among committed values. The security can be based on the Ring-SIS problem, which can in turn be based on worst-case problems on ideal lattices [Mic02,Mic07,PR06,LM06,PR07]. At a high level, in Ring-SIS one is given several uniformly random elements $a_1, \ldots, a_k$ in a ring, and the goal is to find ring elements $x_1, \ldots, x_k$ (not all zero) such that $x_1 a_1 + \cdots + x_k a_k = 0$ under the constraint that the x_i must be "small" (in some appropriate sense that we formally define later).

1.1 Related Work

There are several earlier works in this area: Kawachi et al.'s work on identification schemes [KTX08] presents a string commitment scheme based on the SIS assumption [Ajt96], where one commits to vectors over $\mathbb{Z}_q$. However, the message space is restricted to vectors of small norm; otherwise, the binding property is lost. This restriction causes problems in the applications we are interested in: for instance, if a player wants to prove (efficiently) that he has performed an encryption or decryption operation correctly in a cryptosystem that uses the ring $\mathbb{Z}_q$, one typically requires a commitment scheme that is linearly homomorphic and can commit to *arbitrary* vectors over $\mathbb{Z}_q$ rather that only short ones.

In [JKPT12], Jain et al. proposed a commitment scheme where the hiding property is based on the Learning Parity with Noise (LPN) assumption, a special case of the Learning With Errors (LWE) assumption [Reg05]. They also constructed zero-knowledge proofs to prove general relations on bit strings. A generalization of [JKPT12] was proposed by Xie et al. [XXW13]. Their work presents a commitment scheme that is based on Ring-LWE [LPR10] instead of LPN, and they build Σ-protocols from it. Further Σ-protocols based on (Ring-)LWE

encryption schemes were presented by Asharov et al. [AJL+12] and Benhamouda et al. [BCK+14].

A main drawback of all these previous schemes is that the zero-knowledge proofs had a non-negligible soundness error, and hence one needs many iterations to have full security. In [BKLP15], a commitment scheme, as well as companion zero-knowledge protocols were constructed with much better efficiency: one can commit to a vector over $\mathbb{Z}_q$ resulting in a commitment that is only a constant factor larger than the committed vector. Furthermore, they gave protocols for proving knowledge of a committed string as well as proving linear and multiplicative relations on committed values. These are efficient in the sense that the soundness error is negligible already for a single iteration of the protocol. The commitments are unconditionally binding and computationally hiding, and the underlying assumption is Ring-LWE for a rather constrained parameter set.

Very recently, Lyubashevsky and Seiler [LS17] proposed an explicit construction for certain sets that are also used in our commitment scheme (see Section 3). While their approach mainly focuses on certain sets of parameters, our construction of these can be used in a more general setting.

1.2 Our Contributions

We adopt the following notation: let $R = \mathbb{Z}[\zeta_m] \cong \mathbb{Z}[X]/\langle \Phi_m(X) \rangle$ denote the mth cyclotomic ring, where $\Phi_m(X)$ is the mth cyclotomic polynomial, which has degree $N = \varphi(m)$. Let q be a suitable prime integer and denote $R_q = R/qR \cong \mathbb{F}_q[X]/\langle \Phi_m(X) \rangle$.

A New Commitment Scheme. We propose a commitment scheme that allows committing to an N-vector over $\mathbb{F}_q$, or equivalently, an element in R_q. A commitment consists of just two elements from R_q. The scheme is *statistically hiding* and *computationally binding*, with security based on an instantiation of the Ring-SIS problem. The protocol and analysis uses the *powerful basis* (or "tensored") representation of the ring R [LPR13], rather than the better-known power basis of $\mathbb{Z}[X]/\langle \Phi_m(X) \rangle$ as in previous works. This allows us to obtain a tighter bound, in the *canonical embedding*, on the size of the Ring-SIS solution that we extract from a cheating prover. This in turns leads to a tighter connection to worst-case lattice problems than in previous work, and a broader range of instantiations (e.g., choices of ring). More specifically, the Ring-SIS norm bound is $\tilde{O}(N^{3.5})$.

Zero-Knowledge Protocols. We give (honest-verifier) zero-knowledge protocols for proving knowledge of committed values, and for linear relations among committed values. The protocols achieve negligible soundness error[4] in one iteration, and the communication overhead of a proof, compared to just sending the commitment, is only a logarithmic factor in the security parameter. For the

[4] Already [Lyu09] gave proofs with negligible soundness error, but ours (similar to [BKLP15]) have stronger properties: [Lyu09] extracts a collision in some underlying hash function, while we have to extract a witness for the statement.

soundness analysis we require an exponentially large set of invertible and short elements in R_q. We give a new construction of such a set, defined over a subset of the powerful basis. This is what ultimately allows the improved reduction from Ring-SIS.

Zero-Knowledge Proof of Shortness. We also give a protocol for proving that a committed value is short. This protocol does not have negligible soundness error, but it can be made efficient in an amortized sense using recent work [BDLN16,CDXY17]. We note that [BKLP15] also did not have a one-shot efficient shortness proof.

Threshold Protocols. We show how to use these tools to build maliciously secure threshold key generation and decryption protocols for a class of LWE-based cryptosystems. The communication overhead of the distributed decryption, compared to just sending the ciphertext, is a logarithmic factor in the security parameter, when the cost is amortized over several decryptions, as follows: the parties need to generate some special committed values that do not depend on the ciphertexts to decrypt later. The cost of this is cheap when amortized over many values. Once this is done, the parties can decrypt a number of ciphertexts, and each such decryption will be cheap, even if they must be done one at a time.

In comparison to [BKLP15], which is the most closely related previous work, we have achieved "the other flavor" of commitment, namely, statistically hiding and computationally binding, with similar efficiency but smaller commitment size (2 versus at least 3 ring elements). We obtain "everlasting security," i.e., our computational assumption only has to hold until we stop using the public key. In contrast, when using unconditional binding and computational hiding (as in [BKLP15]) the assumption must hold "forever", because such commitments can be broken offline, at any point after the protocol. In addition, our underlying computational assumption is different and potentially weaker: we use Ring-SIS versus Ring-LWE. This is an advantage because Ring-SIS is at least as hard as a corresponding instantiation of Ring-LWE, and the known worst-case hardness theorems for Ring-SIS are stronger: the reductions are classical, while for Ring-LWE only quantum reductions are known [LPR10,PRS17]. Finally, we instantiate our scheme with a larger class of rings than [BKLP15], namely, arbitrary cyclotomics.

2 Preliminaries

We use $\oplus, \odot$ to denote the coordinate-wise addition and multiplication of two vectors. The tensor product of matrices or rings is denoted by $\otimes$. The statistical security parameter is κ, while λ is the computational security parameter. The set $\{1, ..., n\}$ is denoted by $[n]$.

The Ring-SIS problem, as we use it in our work, is defined using concepts and tools used in [PR07,LPR13], such as the *canonical embedding* and the *powerful basis*. This permits tighter analysis that avoids cruder concepts like ring "expansion factors."

Consider the polynomial $\Phi_m(X)$ whose roots are all the primitive complex mth roots of unity. The polynomial $\Phi_m(X)$ is called mth cyclotomic polynomial and it is a standard fact that $\Phi_m(X) \in \mathbb{Z}[X]$ and that it is monic and irreducible over $\mathbb{Q}$. For $m \geq 2$ the roots are not in $\mathbb{Q}$ so we can consider the number field $K = \mathbb{Q}[X]/\langle \Phi_m(X) \rangle$ and its ring of integers $R = \mathbb{Z}[X]/\langle \Phi_m(X) \rangle$. The extension has degree $N = \deg(\Phi_m(X)) = \varphi(m)$, and K is a $\mathbb{Q}$-vector space with the basis[5] $\{1, \zeta_m, \zeta_m^2, \ldots, \zeta_m^{N-1}\}$, and the same holds for R as a $\mathbb{Z}$-module with the same basis.

Given the factorization $m = m_1 \cdots m_\ell$ into coprime prime powers m_i, we consider the rings of integers for each $K_i = \mathbb{Q}(\zeta_{m_i})$, which are denoted $R_i = \mathbb{Z}[\zeta_{m_i}]$. Then R can be written as the tensor product of these R_i, $R = \bigotimes_{i=1}^{\ell} R_i$. This leads to a different basis of K, R:

Definition 1 ([LPR13]). *The* powerful basis $\boldsymbol{B}$ *of* $K = \mathbb{Q}(\zeta_m), R = \mathbb{Z}[\zeta_m]$ *is defined as follows:*

- *If* m *is a prime power, then* $\boldsymbol{B}$ *is the vector* $(\zeta_m^i)_{0 \leq i < N} \in R$.
- *If* $m = \prod_{i=1}^{\ell} m_i$ *where* m_i, m_j *are coprime prime powers, then* $\boldsymbol{B} = \bigotimes_{i=1}^{\ell} \boldsymbol{B}_i$ *where* $\boldsymbol{B}_i$ *is the powerful basis of* R_i.

For a $\mathbb{Z}$-basis $\boldsymbol{B} = \{b_1, ..., b_N\}$ of the field K we define the ℓ_2, ℓ_∞-norm of $x = x_1 b_1 + \cdots + x_N b_N$ as $||x||_2 = (\sum_i |x_i|^2)^{1/2}$ and $||x||_\infty = \max_i |x_i|$

The Canonical Embedding. The canonical embedding of K into $\mathbb{C}^N$ is defined as follows: let ζ_m be the mth primitive root of unity in K and $\{\omega_m^i\}_{i \in \mathbb{Z}_m^*}$ be all N of the mth primitive roots of unity in $\mathbb{C}$. Then σ_i is the unique ring homomorphism that fixes $\mathbb{Q}$ and maps ζ_m to ω_m^i, for $i \in \mathbb{Z}_m^*$. Concatenating all these σ_i yields the injective ring homomorphism (where the ring operations are coordinate-wise)

$$\begin{aligned} \sigma \colon K &\to \mathbb{C}^N \\ x &\mapsto (\sigma_i(x))_{i \in \mathbb{Z}_m^*}. \end{aligned}$$

For $x \in K$ we define its *canonical norms* similarly as above, namely we let $||x||_2^c = (\sum_i |\sigma_i(x)|^2)^{1/2}$ and $||x||_\infty^c = \max_i |\sigma_i(x)|$. The canonical embedding σ has the advantage that both addition and multiplication of field elements is coordinate-wise under it, i.e., $\forall x, y \in K$, $\sigma(x + y) = \sigma(x) \oplus \sigma(y)$ and $\sigma(x \cdot y) = \sigma(x) \odot \sigma(y)$. This gives tight bounds on products[6] of elements of K:

$$\forall x, y \in K : \ ||x \cdot y||_2^c \leq ||x||_\infty^c \cdot ||y||_2^c$$

By [LPR13, Lemma 4.3], we have the following bounds on the relationship between $||\cdot||_p^c$-norms and their $||\cdot||_p$-counterparts.

[5] This basis is normally referred to as the power basis.

[6] This actually holds for arbitrary ℓ_p norms, not just the ℓ_2 norm.

Remark 1. We consider elements of K with respect to the powerful basis $\boldsymbol{B}$. Let $x \in K, \boldsymbol{y} \in K^k$ for $k \in \mathbb{N}$. Then

1. $||x||_\infty^c \leq \sqrt{N} \cdot ||x||_2$ and $||\boldsymbol{y}||_\infty^c \leq \sqrt{N} \cdot ||\boldsymbol{y}||_2$
2. $||x||_\infty^c \leq N \cdot ||x||_\infty$ and $||\boldsymbol{y}||_\infty^c \leq N \cdot k \cdot ||\boldsymbol{y}||_\infty$
3. $||x||_2^c \leq \sqrt{m} \cdot ||x||_2$ and $||\boldsymbol{y}||_2^c \leq \sqrt{m \cdot k} \cdot ||\boldsymbol{y}||_2$

The Ring-SIS Problem. Letting q be an arbitrary prime integer, we define $R_q = \mathbb{Z}[X]/\langle \Phi_m(X), q \rangle$. The algebraic properties of this ring depend upon the splitting of the ideal $\langle q \rangle$ in R. The powerful $\mathbb{Z}$-basis of R, and its reduction mod qR, are both $\mathbb{Z}_q$-bases of R_q. Naturally, for any $x \in R$ the powerful-basis coefficients of x mod qR are just the coefficients of x, all reduced modulo q. Conversely, to obtain the distinguished R-representative of any $x \in R_q$, we simply lift the $\mathbb{Z}_q$-coefficients with respect to the powerful basis to their distinguished $\mathbb{Z}$-representatives.

We can now define the *Ring-SIS* problem, using the canonical embedding to measure the norm of the solution (the choice of which yields tighter connections to worst-case problems on ideal lattices in the ring [PR07]).

Definition 2. *Let $m, q \in \mathbb{N}^+$ with q a prime not dividing m, and let R, R_q be defined as above. For a bound $u \in \mathbb{R}$ and a parameter $k \in \mathbb{N}^+$, the (inhomogeneous)* Ring-SIS *problem is: given uniformly random and independent $a_1, \ldots, a_k, t \in R_q$, find $\boldsymbol{x} \in R^k$ such that*

$$||\boldsymbol{x}||_2^c \leq u \text{ and } \sum_{i=1}^{k} a_i \cdot \boldsymbol{x}[i] = t.$$

Normal Distributions. The continuous normal distribution over $\mathbb{R}^d$ centered at $\boldsymbol{v} \in \mathbb{R}^d$ with standard deviation σ has probability density function

$$\rho_{\boldsymbol{v},\sigma}^d(\boldsymbol{x}) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(\frac{-||\boldsymbol{x} - \boldsymbol{v}||_2^2}{2\sigma^2}\right)$$

In our zero-knowledge proofs we will use a discrete version. The *discrete normal distribution* over R^k centered at $\boldsymbol{v} \in R^k$ with standard deviation σ is given by the distribution function (for all $\boldsymbol{x} \in R^k$)

$$\mathcal{N}_{\boldsymbol{v},\sigma}^k(\boldsymbol{x}) = \rho_{\boldsymbol{v},\sigma}^{k \cdot N}(\boldsymbol{x}) / \rho_\sigma^{k \cdot N}(R^k),$$

where the norm is $|| \cdot ||_2$ (relative to the powerful basis) and where we omit the subscript $\boldsymbol{v}$ when it is zero. As we will use specific properties of these distributions only in the proofs in the appendix, we refer the reader to Appendix C for more information.

3 Short and Invertible Elements

Our commitment scheme and its zero-knowledge proofs rely on specific properties of R_q. In the construction we will need to make use of an (exponentially) large set of elements $a \in R$ that are short, but invertible as $\overline{a} \in R_q$.

Definition 3 (Commitment-friendly Set). *Let R, R_q be defined as before, and let $\boldsymbol{B}, \overline{\boldsymbol{B}}$ be the powerful bases of R, R_q. The set $D \subseteq \boldsymbol{B}$ is called* commitment-friendly *if every non-trivial $\mathbb{F}_q$-combination of its corresponding elements in $\overline{\boldsymbol{B}}$ is invertible in R_q.*

The constructions in this section will be presented without proofs, which are deferred to Appendix A. We give an explicit subset of the canonical basis which is commitment-friendly. Let $\mathrm{ord}_m(q)$ be the multiplicative order of q mod m.

Lemma 1. *Let $m = m_1 \cdots m_\ell$ be the unique factorization of m into prime powers, let moreover $d_1 = \mathrm{ord}_{m_1}(q)$ and $d_i = \mathrm{ord}_{m_1 \cdots m_i}(q) / \mathrm{ord}_{m_1 \cdots m_{i-1}}(q)$. Let ζ_{m_i} be the m_i-th primitive root of unity in the algebraic closure $\overline{\mathbb{Q}}$. Consider the set*

$$\mathsf{E} = \{1, \zeta_1, \ldots, \zeta_1^{d_1 - 1}\} \otimes \cdots \otimes \{1, \zeta_\ell, \ldots, \zeta_\ell^{d_\ell - 1}\}.$$

Then E is commitment-friendly and has maximal size.

Using the powerful basis we are able to tightly control the canonical norms of ring elements and products of them. But for technical reasons, we also need to bound the $||\cdot||_\infty$-norm on products in R in some cases. We define a special set $D' \subset D$, that we call *multiplication-friendly*. To ease readability, for elements $x, y \in R$ that respectively are integer combinations of the basis subsets D, D' we write $x \in D, y \in D'$ (but note that still $D, D' \subseteq \boldsymbol{B}$). Moreover, define $S_{D,\beta}$ to be the set of those $x \in D$ such that $||x||_\infty \leq \beta$.

Definition 4. *Let R, R_q be defined as before. Let $D \subseteq \boldsymbol{B}$ be a commitment-friendly set. Then $D' \subset D$ is called* (s,t)-multiplication-friendly *if for all $x, y \in D'$ and $z \in R$:*

1. *If $||x||_\infty \leq r_1, ||z||_p^c \leq r_2$ then $||x \cdot z||_p^c \leq s \cdot r_1 \cdot r_2$ for any ℓ_p-norm.*
2. *If $||x||_\infty \leq r_1, ||y||_\infty \leq r_2$ then $||x \cdot y||_2 \leq t \cdot r_1 \cdot r_2$.*

Intuitively, a set D' is multiplication-friendly if we can give a tight upper bound on the norm of the product of its elements, both for the canonical and the coefficient norm. The latter is generally complicated, as it involves reductions modulo $\Phi_m(X)$ in R_q.

Proposition 1. $\mathsf{E}' = \{1, \zeta_1, \ldots, \zeta_1^{\lfloor \frac{d_1 - 1}{2} \rfloor}\} \otimes \cdots \otimes \{1, \zeta_\ell, \ldots, \zeta_\ell^{\lfloor \frac{d_\ell - 1}{2} \rfloor}\}$
is $(|\mathsf{E}|/2, |\mathsf{E}|^{1.5}/2)$-multiplication-friendly.

4 The Commitment Scheme

In this section, we will present our commitment scheme and prove its security. On a very high level it is related to a previous scheme due to Damgård et al. [DPP93] which can be based on any collision intractable hash function h. To commit to a bit string x, one chooses a random (sufficiently long) string r, and the commitment is defined as $(h(r), \phi, \phi(r) \oplus x)$, where ϕ is a universal hash function. By collision intractability, the committer cannot change his mind about r and hence not about x either. It is hiding by the randomness extraction property of ϕ: if r is long enough compared to $h(r)$, then $\phi(r)$ is essentially uniform and masks x.

The "Ring-SIS function" (see Definition 2) for key $\boldsymbol{a} \in R_q^k$ sends a short vector $\boldsymbol{r} \in R^k$ to $\boldsymbol{a} \cdot \boldsymbol{r} \in R_q$. This can be thought of as a collision intractable function for the right parameters [PR06,LM06,PR07]. In addition, the same type of function can also be used as a randomness extractor by choosing suitable parameters. The intuition behind our scheme is therefore to instantiate the idea from [DPP93] using instances of the Ring-SIS function over R_q for both h and ϕ.

However, in contrast to standard instantiations of [DPP93], both functions are defined over the same polynomial ring and this gives the scheme some nice algebraic properties. It turns out that we can use these for constructing efficient zero-knowledge protocols for the scheme, based on the commitment-friendly and multiplication-friendly sets from the previous section.

Parameter	Explanation
λ, κ	Computational/statistical security parameter
R	The ring over which we define the norms of vectors
R_q	The ring over which we do most of the computations
m	Order of the root of unity used to define R_q, R
q	Prime modulus defining R_q
N	Degree of Φ_m, $N = \varphi(m)$
k	Dimension (over R_q) for the Ring-SIS problem
D	The commitment-friendly set of R
D'	The multiplication-friendly set of R, used to open commitments
s, t	Parameters of the (s, t)-multiplication-friendly set D'
β	Norm bound for honest prover's randomness in ℓ_∞-norm
$\mathcal{D}$	Distribution of honest prover's randomness for commitments
$S_{D,\beta}$	Set of all elements $x \in D \subseteq R$ with ℓ_∞-norm at most β
$\boldsymbol{A}$	Public matrix from $R_q^{2 \times k}$

Fig. 1. Overview of Parameters and Notation.

4.1 Efficient Ring-SIS Commitments

We now describe the commitment scheme we propose in its most general form. It borrows an idea from [BKLP15] for relaxing the condition for a valid opening of a commitment to achieve a better soundness error probability of the Σ-protocols. Fig. 1 gives an overview of the parameters and variables.

CKeyGen: The public commitment key is the specification of a ring R_q with the commitment-friendly set D from Definition 3 and a (s,t)-multiplication-friendly set D' as defined in Definition 4, a uniformly random matrix $\boldsymbol{A} \in R_q^{2\times k}$ and a constant β such that $16 \cdot s \cdot t \cdot k \cdot m \cdot \log(k \cdot N) \cdot \beta^3 < q/2$. Finally, define the distribution $\mathcal{D}$ which outputs a $\boldsymbol{v} \in S^k_{D',\beta}$ uniformly at random.

Commit: To commit to a message $x \in R_q$, draw a $\boldsymbol{r} \xleftarrow{\$} \mathcal{D}$ and compute

$$Com(x;\boldsymbol{r}) := \boldsymbol{A}\boldsymbol{r} + \binom{0}{x}.$$

Open: A valid opening of a commitment $\boldsymbol{c}$ is a 3-tuple: $x \in R_q$, $\boldsymbol{r} \in R^k$ and $\mathsf{f} \in S_{D',2\cdot\beta}$. The verifier checks that

$$\boldsymbol{A}\boldsymbol{r} + \binom{0}{\mathsf{f}x} = \mathsf{f}\boldsymbol{c},$$

and that $||\boldsymbol{r}||_2^c \leq 4 \cdot t \cdot k \cdot m \cdot \log(k \cdot N) \cdot \beta^2$.

We will often omit the choice of randomness and write $C(x)$ or $C(x;\boldsymbol{r})$ instead of $Com(x;\boldsymbol{r})$. Note that an honest committer can always open by letting $\mathsf{f} = 1$, and would always have its value of $\boldsymbol{r}$ be shorter than $4 \cdot t \cdot k \cdot m \cdot \log(k \cdot N) \cdot \beta^2$, namely it would have $||\cdot||_2^c$-norm at most $\sqrt{km} \cdot \beta$. We only allow for these relaxed conditions in order to get soundness and zero-knowledge for the protocols we propose in Section 5: we will only be able to guarantee that a dishonest prover can open his values using f-values that are (possibly) not 1, and $\boldsymbol{r}$-values of norm larger than β. This is fine, as long as the scheme is still binding under the relaxed condition, as indeed we show below.

The commitment scheme can be extended to allow committing to vectors $\boldsymbol{x} \in R_q^d$ for fixed d. Appendix B shows the necessary modifications of the commitment scheme and proofs for the hiding and binding property.

4.2 Security

We will now prove the security guarantees of our commitment scheme. The first lemma shows that breaking the binding property implies one can solve the Ring-SIS problem over R_q. The second lemma shows that the commitment scheme is statistically hiding.

Lemma 2 (Binding Property). *From a commitment $\boldsymbol{c}$ and correct openings $\boldsymbol{r}, \mathsf{f}, \boldsymbol{r}', \mathsf{f}'$ to two different messages x, x', one can efficiently compute a solution with norm $16 \cdot s \cdot t \cdot k \cdot m \cdot \log(kN) \cdot \beta^3$ to the Ring-SIS problem instance defined by the top row of $\boldsymbol{A}$.*

Proof. Let $\boldsymbol{c}$ and x, $\boldsymbol{r}$, f and x', $\boldsymbol{r}'$, f' be as assumed in the lemma. Then

$$\boldsymbol{A}(f'\boldsymbol{r}) + \begin{pmatrix} 0 \\ ff'x \end{pmatrix} = ff'\boldsymbol{c} = \boldsymbol{A}(f\boldsymbol{r}') + \begin{pmatrix} 0 \\ ff'x' \end{pmatrix}$$

and so

$$\boldsymbol{A}(f'\boldsymbol{r} - f\boldsymbol{r}') = \begin{pmatrix} 0 \\ ff'(x-x') \end{pmatrix}.$$

Since $x - x' \neq 0$ and the actions of both f, f' are invertible, we have $ff'(x-x') \neq 0$. Then it must be that also $f'\boldsymbol{r} - f\boldsymbol{r'} \neq 0$ since otherwise the above equation would be false. Hence we have found a solution $f'\boldsymbol{r} - f\boldsymbol{r'}$ to the Ring-SIS instance defined by the top row of $\boldsymbol{A}$. By Definition 4, we can bound $||f'\boldsymbol{r} - f\boldsymbol{r'}||_\infty^c \leq 2 \cdot ||f'\boldsymbol{r}||_2^c$ as

$$||f'\boldsymbol{r} - f\boldsymbol{r}'||_2^c \leq 2s \cdot (2 \cdot \beta) \cdot (4t \cdot k \cdot m \cdot \log(k \cdot N) \cdot \beta^2). \qquad \square$$

In the proof of the hiding property we need that the multiplication with the matrix $\boldsymbol{A}$ is a universal hash function. Computing $\boldsymbol{Ar}$ consists of evaluating functions of the form $f_{\boldsymbol{a}}(\boldsymbol{r}) = \langle \boldsymbol{a}, \boldsymbol{r} \rangle$.

Proposition 2. *The function*

$$\begin{aligned} f_{\boldsymbol{a}}(\boldsymbol{r}) : R_q^k \times S_{D',\beta}^k &\to R_q \\ (\boldsymbol{a}, \boldsymbol{r}) &\mapsto \langle \boldsymbol{a}, \boldsymbol{r} \rangle \end{aligned}$$

is a universal hash function.

Proof. Due to the direct product structure of R_q, we can think of the function $f_{\boldsymbol{a}}(\cdot)$ as a direct product of u functions defined over $\mathbb{F}_i = \mathbb{F}_q[X]/\langle f_i(X) \rangle$. Each of these functions is universal since they compute the dot product over fields and so they have collision probability $1/|\mathbb{F}_i|$. Now since $f_{\boldsymbol{a}}$ is linear, a collision occurs if and only if the function sends a non-zero input to 0. However, a non-zero vector in $\boldsymbol{r} \in S_{D',\beta}^k$ is also non-zero when reduced modulo any $f_i(X)$ due to the invertibility of each value in D'. Hence a collision only occurs if one occurs in each subfield, and so the collision probability is $\prod_i 1/|\mathbb{F}_i| = 1/|R_q|$. $\square$

Lemma 3 (Hiding Property). *Assume the distribution $\mathcal{D}$ and R_q, k are chosen such that 1) the min-entropy of a vector drawn from $\mathcal{D}$ is at least $2\log(|R_q|) + \kappa$ where κ is a (statistical) security parameter, and 2) the class of functions $\{f_{\boldsymbol{a}} \mid \boldsymbol{a} \in R_q^k\}$ where $f_{\boldsymbol{a}}(\boldsymbol{r}) = \boldsymbol{a} \cdot \boldsymbol{r}$ is universal when mapping the support of $\mathcal{D}$ to R_q. Then the scheme is statistically hiding.*

Proof. Note that a commitment gives the adversary $\log(|R_q|)$ bits of information on $\boldsymbol{r}$, namely the dot product of $\boldsymbol{r}$ with the top row $\boldsymbol{a}_0$ of $\boldsymbol{A}$. So even given this dot-product we have $\log(|R_q|) + \kappa$ bits of randomness left in $\boldsymbol{r}$. Let $\boldsymbol{a}_1$ be the bottom row of $\boldsymbol{A}$. Then from the assumptions and the left-over hash lemma, it follows that $f_{\boldsymbol{a}_1}(\boldsymbol{r})$ is statistically close to random, even given $\boldsymbol{a}_0 \cdot \boldsymbol{r}$ and so the scheme is indeed statistically hiding. $\square$

Instantiating the Scheme. We now instantiate the commitment scheme with the commitment-friendly set E and the $(|\mathsf{E}|/2, |\mathsf{E}|^{1.5}/2)$-multiplication-friendly set E'. By Lemma 2 the scheme is binding as long as

$$||r||_2^c \leq 2 \cdot |\mathsf{E}|^{3/2} \cdot m \cdot k \cdot \log(k \cdot N) \cdot \beta^2$$

and the reduction outputs a solution to the Ring-SIS instance with $||\cdot||_2^c$-norm at most $4 \cdot |\mathsf{E}|^{5/2} \cdot m \cdot k \cdot \log(k \cdot N) \cdot \beta^3$. This yields a Ring-SIS solution of size $O(N^{3.5} \cdot \log^2(N))$ in the canonical embedding.

To get an efficient scheme in practice, one would set $m = O(\lambda)$ and $q = O(\lambda^c)$ for some constant c. We want to choose $\beta = O(1)$ (which implies $k = \Theta(\log \lambda)$ to obtain a hash function with collisions) to get a bound as tight as possible. In Appendix A.3 we show that one can achieve $|\mathsf{E}|, |\mathsf{E}'| = \Theta(\lambda)$, so the scheme will be hiding according to Lemma 3 even for β of constant size:

Remark 2. Let $|\mathsf{E}'| = \Omega(\lambda)$, then under the above conditions, the commitment is hiding for $\beta = O(1)$.

Proof. We choose $N = O(\lambda)$ and $\boldsymbol{r}$ has entropy $\log |\mathcal{D}|$ as the coefficients are chosen uniformly at random. Therefore

$$|\mathsf{E}'| \cdot k \cdot \log \beta \geq 3 \cdot N \cdot \log q > 2 \cdot N \cdot \log q + \lambda$$

and such a constant β exists. □

5 Zero-Knowledge Proofs

In this section, we describe Σ-protocols that can be constructed for our commitment scheme. The protocols use rejection sampling which was introduced in the context of lattice-based constructions in [Lyu08,Lyu09,Lyu12]. Rejection sampling allows to hide the randomness during the proof. The protocols use an auxiliary commitment scheme C_{aux} during the first round for technical reasons - here we can use our own commitment scheme, but use the notation C_{aux} to make the presentation clearer.

Our protocols have slightly weaker properties than usually considered for Σ-protocols, but this does not affect their usefulness in practice: we get statistical honest verifier zero-knowledge rather than perfect, and we get computational soundness rather than perfect, in the sense that a prover who can answer two different challenges must either know the witness we want him to know, or he can break the binding property of C_{aux}. All protocols, except the one for proving bounds (Π_{BOUND}), have soundness error $1/\beta^{|D'|}$, which we show to be negligible (for certain parameter sets) in Appendix A.3. In Appendix C.5 we discuss ways in which we can make our protocols be zero-knowledge against a dishonest verifier[7].

The communication complexity of the proofs is dominated by the size of the prover's last message, which will be $O(kN \log(q))$ bits. A commitment is of

[7] Note that this is not obvious (even using rewinding), when the challenge space is large.

size $O(N\log(q))$ bits and we in practice choose $k = O(\log(\lambda))$, so running the protocols adds very little asymptotic overhead.

In the commitment scheme, the prover chooses the randomness $\boldsymbol{r}$ from $\mathcal{D}$, while the challenger chooses $\mathsf{d} \in D'$. $T = \sqrt{k} \cdot t \cdot \beta^2$ then is the bound on the l_2-norm of $\mathsf{d}\cdot\boldsymbol{r}$ for $\mathsf{d} \in S_{D',\beta}, \boldsymbol{r} \in S^k_{D',\beta}$. This bound can be obtained from Definition 4 as D' is (s,t)-multiplication-friendly. We moreover set $\sigma = T \cdot \log(kN)$ as the variance used in sampling randomness in the protocols, and $M = O(1)$ to be some constant. We elaborate on this choice in Appendix C.

5.1 Proof for Opening a Commitment

Suppose the prover has published $\boldsymbol{c} = C(x;\boldsymbol{r})$ and claims to know a valid opening. Then consider the following protocol to prove this:

Protocol $\Pi_{\mathbf{Open}}$

1. The prover chooses $\mu \xleftarrow{\$} R_q$ and $\boldsymbol{\rho} \xleftarrow{\$} \mathcal{N}^k_\sigma$, computes $\boldsymbol{t} = C(\mu;\boldsymbol{\rho})$ and sends $\boldsymbol{c}_{aux} = C_{aux}(\boldsymbol{t})$ to the verifier.
2. The verifier sends a random challenge $\mathsf{d} \xleftarrow{\$} S_{D',\beta}$.
3. The prover first checks that $\mathsf{d} \in S_{D',\beta}$. He then computes $z = \mu + \mathsf{d}x$, $\boldsymbol{r}_z = \boldsymbol{\rho} + \mathsf{d}\boldsymbol{r}$. The prover either aborts with probability $1 - \min\left(1, \frac{\mathcal{N}^k_\sigma(\boldsymbol{r}_z)}{M\mathcal{N}^k_{\mathsf{d}r,\sigma}(\boldsymbol{r}_z)}\right)$. or sends $z, \boldsymbol{r}_z$ and opening information u_{aux} for $\boldsymbol{c}_{aux}$ to the verifier.
4. The verifier checks that u_{aux} is valid, that $C(z;\boldsymbol{z}_r) = \boldsymbol{t} + \mathsf{d}\boldsymbol{c}$, and that $||\boldsymbol{r}_z||_2 \leq 2\sigma\sqrt{kN}$.

We now look at the properties of this protocol. The proof can be found in Appendix C.1.

Lemma 4. *The protocol Π_{Open} has the following properties:*

- Completeness: *The verifier accepts with overwhelming probability when Π_{Open} does not abort. The probability of abort is at most $1 - \frac{1-2^{-100}}{M}$.*
- Special Soundness: *Given a commitment $\boldsymbol{c}$ and a pair of transcripts for Π_{Open} $(\boldsymbol{c}_{aux}, \mathsf{d}, (u_{aux}, t, z, \boldsymbol{r}_z))$, $(\boldsymbol{c}_{aux}, \mathsf{d}', (u'_{aux}, t', z', \boldsymbol{r}'_z))$ where $\mathsf{d} \neq \mathsf{d}'$, we can extract either a witness for breaking the auxiliary commitment scheme, or a valid opening $(x, \boldsymbol{r}, \mathsf{f})$ of $\boldsymbol{c}$ with $||\boldsymbol{r}||_2 \leq 4\sigma\sqrt{kN}, ||\mathsf{f}||_\infty \leq 2\cdot\beta$.*
- Honest-Verifier Zero-Knowledge: *Transcripts of Π_{Open} with an honest verifier can be simulated with statistically indistinguishable distribution.*

Proof for Opening to a Specific Message. The protocol Π_{Open} demonstrates that the prover knows how to open a commitment, without revealing either the randomness or the message. An easy variant, which we will call $\Pi_{\text{Open-x}}$, can be used to show that the prover can open $\boldsymbol{c}$ to a specific message x: it is enough to show that a commitment can be opened to 0, since one can use that protocol on input $\boldsymbol{c} - C(x;\boldsymbol{0})$. Now, to prove that a commitment can be opened to 0, execute Π_{Open} where $\mu = 0$. As a result, $z = 0$ and the verifier checks that this is indeed the case. It trivial to show completeness, special soundness and honest verifier zero-knowledge for this protocol, and we leave this to the reader.

Proof for Linear Relation. Suppose that the prover has published two commitments $\boldsymbol{c}_1 = C(x_1; \boldsymbol{r}_1), \boldsymbol{c}_2 = C(x_2; \boldsymbol{r}_2)$ and claims that $x_2 = g(x_1)$ for a linear function g. The protocol Π_{LIN} for proving this relation is similar to Π_{OPEN}, but the prover's first message contains two commitments to vectors that are linearly related by g. The protocol as well as its properties can be found in Appendix C.2.

Proof for Sum. Suppose that the prover has published three commitments $\boldsymbol{c}_1 = C(x_1; \boldsymbol{r}_1)$, $\boldsymbol{c}_2 = C(x_2; \boldsymbol{r}_2)$, $\boldsymbol{c}_3 = C(x_3; \boldsymbol{r}_3)$ and claims that $x_3 = \alpha_1 x_1 + \alpha_2 x_2$ where $\alpha_1, \alpha_2 \in R_q$ are public constants. The protocol Π_{SUM} is similar to the previous protocol. It's specification and properties are in Appendix C.3.

Proving Bounds. Suppose that the prover has published a commitment $\boldsymbol{c} = C(x; \boldsymbol{r}_x)$ and claims that the norm of x is small. The idea is to add a short random value μ to x and check whether the sum is sufficiently short. We can only allow for small challenges, i.e. we restrict the challenge space here to $\{0, 1\}$. The protocol can be made efficient in an amortized sense using [BDLN16,CDXY17]. The protocol Π_{BOUND} and its properties can be found in Appendix C.4.

6 Actively Secure Threshold Protocols

In this section, we describe how to efficiently compile passively secure threshold protocols for key generation and decryption into their actively secure counterpart for n parties $\mathcal{P} = \{P_1, \ldots, P_n\}$, out of which at most $n-1$ can be malicious ($\mathcal{I} \subset \mathcal{P}$ denotes the corrupted parties). Our transformation is generic and applies to schemes that are based on the (Ring-)LWE assumption. Due to space constraints, we only present the key generation here, whereas the decryption protocol can be found in Appendix D.

6.1 Defining Threshold Cryptosystems

We start with an abstract definition of the cryptosystems to which our solution applies: let $d, l_c, l_{\mathsf{pk}}, l_{\mathsf{sk}}, l_s, l_r, l_d, \beta_s, \omega_s, \beta_d, \omega_d \in \mathbb{N}, \beta_s, \beta_d \ll q$ (we assume that these parameters implicitly are functions of λ).

Let us make the simplifying assumption[8] that $l_c, l_{\mathsf{pk}}, l_{\mathsf{sk}}, l_s, l_r, l_d$ are multiples of N. Let $\mathcal{U}_\beta^\ell$ be an algorithm that efficiently samples from $\mathbb{F}_q^\ell$ by choosing each coordinate uniformly at random from $[-\beta, \beta]$ (when representing each $\mathbb{F}_q$-element by its representative from $(-q/2, q/2]$).

The probabilistic encryption algorithm Enc maps a string $m \in \{0,1\}^d$ to an element $c \in \mathbb{F}_q^{l_c}$. Moreover, we define generic algorithms $\mathsf{KG}, \mathsf{Dec}$ for key generation and decryption. These depend on matrices $\boldsymbol{F}_a^{\mathsf{KG}} \in \mathbb{F}_q^{(l_{\mathsf{pk}}+l_{\mathsf{sk}}) \times l_s}$ for the

[8] One might also use larger blocks for efficiency reasons, as our commitment scheme supports to commit to R_q-vectors. See Appendix B for details.

key generation and $\boldsymbol{F}_r \in \mathbb{F}_q^{l_d \times l_r}$,$\boldsymbol{F}_c \in \mathbb{F}_q^{l_d \times l_c}$ for the decryption. These matrices[9] are implicitly defined by a CRS and the ciphertext c. The decryption additionally uses a publicly known algorithm

$$\mathsf{decode} : \mathbb{F}_q^{l_d} \to \{0,1\}^d \cup \{\perp\}$$

that removes the noise in the ciphertext and differs depending on whether the message is stored in the higher or lower bits of the ciphertext. We define the key generation and decryption abstractly as being "mostly linear", i.e. both operations consist of multiplying a secret vector with a known public matrix, plus eventually adding some noise. The algorithms KG and Dec are defined as follows:

$\mathsf{KG}(1^\lambda, n, \boldsymbol{F}_a^{\mathsf{KG}}, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_n)$**:**

1. For $i \in [n]$ compute $(\mathsf{pk}_i, \mathsf{sk}_i) = \boldsymbol{F}_a^{\mathsf{KG}} \boldsymbol{s}_i$.
2. Output $(\mathsf{pk} = \sum_i \mathsf{pk}_i, \mathsf{pk}_1, \ldots, \mathsf{pk}_n, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$.

$\mathsf{Dec}(\boldsymbol{F}_c, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_n)$**:**

1. For $i \in [n]$ compute $\boldsymbol{d}_i = \boldsymbol{F}_r \boldsymbol{e}_i + \boldsymbol{F}_c \mathsf{sk}_i$.
2. Output $(\mathsf{decode}(\sum_i \boldsymbol{d}_i), \boldsymbol{d}_1, \ldots, \boldsymbol{d}_n)$.

Definition 5 (Distributed Cryptosystem). *The tuple of (probabilistic) polynomial-time algorithms $D = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ is a distributed cryptosystem if there exist protocols $\Pi_{\mathrm{KG}}, \Pi_{\mathrm{DEC}}$ that securely implement $\mathcal{F}_{\mathrm{KGD}}$ (Fig. 2).*

The parameters ω_s, ω_d in Fig. 2 allow the adversary in the malicious setting to choose slightly larger values than in the semi-honest case. This is necessary because Π_{BOUND} naturally comes with some tightness slack. Our above definition captures the encryption schemes [BV11,BGV12] directly, but can also be adapted to [FV12] with minor changes to the Dec procedure. Unfortunately it does not directly apply to [HPS98] due to the structure of pk, but to the more efficient, recently proposed Learning-with-Rounding scheme [CKLS16] due to Cheon et al. In these cases $\mathcal{F}_{\mathrm{KGD}}$ can easily be implemented with passive security. For security against active adversaries, one has to ensure that $\boldsymbol{s}_i, \boldsymbol{e}_i$ are bounded as in $\mathcal{F}_{\mathrm{KGD}}$. Moreover, $\mathsf{pk}_i, \mathsf{sk}_i, \boldsymbol{d}_i$ of the dishonest parties may depend on those values of the honest parties or not be computed using $\boldsymbol{F}_a^{\mathsf{KG}}, \boldsymbol{F}_r, \boldsymbol{F}_c$ at all.

6.2 Actively Secure Key Generation

The key generation protocol can informally be described as follows: in a first step, all parties sample a value $\boldsymbol{s}_i$ that they commit to. They then prove in zero-knowledge that this commitment indeed contains a short value. We moreover let each party commit to the values $\mathsf{pk}_i, \mathsf{sk}_i$ which can be computed from $\boldsymbol{s}_i$ and let them prove that the commitments can indeed be obtained using the public linear transform $\boldsymbol{F}_a^{\mathsf{KG}}$ using our zero-knowledge proofs. Finally we let the parties open pk_i, so that they can then individually compute the public key.

[9] We may also just sample $\boldsymbol{F}_a^{\mathsf{KG}}$ using a distributed coin-flipping protocol using our commitment scheme.

Functionality $\mathcal{F}_{\mathbf{KGD}}$

Key Generation:

1. Wait for each party P_i to input $\left(\mathsf{KeyGen}, \boldsymbol{F}_a^{\mathsf{KG}}\right)$.
2. For each $P_i \in \mathcal{I}$, $\mathcal{A}$ inputs $\boldsymbol{s}_i$. If $\boldsymbol{s}_i \notin \mathbb{F}_q^{l_s}$ or $||\boldsymbol{s}_i||_\infty > \omega_s \cdot \beta_s$, then output (Abort, P_i) to all honest parties and stop.
3. For each $P_i \in \mathcal{P} \setminus \mathcal{I}$ sample $\boldsymbol{s}_i \xleftarrow{\$} \mathcal{U}_{\beta_s}^{l_s}$.
4. Compute $(\mathsf{pk}, \mathsf{pk}_1, \ldots, \mathsf{pk}_n, \mathsf{sk}_1, \ldots, \mathsf{sk}_n) \leftarrow \mathsf{KG}(1^\lambda, n, \boldsymbol{F}_a^{\mathsf{KG}}, \boldsymbol{s}_1, \ldots, \boldsymbol{s}_n)$.
5. Locally store $(\mathsf{Keys}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$ if no such pk has been stored before.
6. Output $\left(\mathsf{pk}, (\mathsf{pk}_i)_{P_i \in \mathcal{P} \setminus \mathcal{I}}\right)$ to $\mathcal{A}$ and $(\mathsf{pk}, \mathsf{sk}_i)$ to each honest P_i.

Decryption:

1. Wait for each party P_i to input $(\mathsf{Decrypt}, \mathsf{pk}, \boldsymbol{F}_c)$.
2. Load $(\mathsf{Keys}, \mathsf{pk}, \mathsf{sk}_1, \ldots, \mathsf{sk}_n)$. If no such entry can be found, abort.
3. For each $P_i \in \mathcal{I}$ $\mathcal{A}$ inputs $\boldsymbol{e}_i$. If $\boldsymbol{e}_i \notin \mathbb{F}_q^{l_r}$ or $||\boldsymbol{e}_i||_\infty > \omega_d \cdot \beta_d$ then output (Abort, P_i) to all honest parties and stop.
4. For each $i \in \mathcal{P} \setminus \mathcal{I}$ sample $\boldsymbol{e}_i \xleftarrow{\$} \mathcal{U}_{\beta_d}^{l_r}$.
5. Compute $(m, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_n) \leftarrow \mathsf{Dec}(\boldsymbol{F}_c, \mathsf{sk}_1, \ldots, \mathsf{sk}_n, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_n)$.
6. Output $\left(m, (\boldsymbol{d}_i)_{P_i \in \mathcal{P} \setminus \mathcal{I}}\right)$ to each dishonest P_i and m to each honest P_i.

Fig. 2. $\mathcal{F}_{\text{KGD}}$: Ideal functionality for distributed key generation.

Protocol $\Pi_{\mathbf{KG}}$

1. Each P_i locally samples $\boldsymbol{s}_i \xleftarrow{\$} \mathcal{U}_{\beta_s}^{l_s}$.
2. Each P_i computes and broadcasts the commitments $C(\boldsymbol{s}_i), C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_i), C(\boldsymbol{F}_a^{\mathsf{sk}}\boldsymbol{s}_i)$.
3. Each P_i uses the following proofs towards all parties. Sample the challenge using $\mathcal{F}_{\text{RAND}}$:
   (a) Π_{BOUND} on $C(\boldsymbol{s}_i)$ to show that $||\boldsymbol{s}_i||_\infty \leq \beta_s$.
   (b) Π_{LIN} on $C(\boldsymbol{s}_i)$, $C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_i)$ using $\boldsymbol{F}_a^{\mathsf{pk}}$.
   (c) Π_{LIN} on $C(\boldsymbol{s}_i)$, $C(\boldsymbol{F}_a^{\mathsf{sk}}\boldsymbol{s}_i)$ using $\boldsymbol{F}_a^{\mathsf{sk}}$.
   If one of the proofs fails then abort.
4. Denote with pk_i the committed value in $C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_i)$. Each P_i proves to all parties that $C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_i)$ contains pk_i using $\Pi_{\text{OPEN-X}}$. If one of the proofs fails, then abort.
5. If all proofs were correct, then output $\mathsf{pk} = \sum_{i \in [n]} \mathsf{pk}_i$.

Fig. 3. Π_{KG}: Protocol for actively secure key generation.

To ease notation, we can write $\boldsymbol{F}_a^{\mathsf{KG}}$ as $\boldsymbol{F}_a^{\mathsf{KG}} = \left(\boldsymbol{F}_a^{\mathsf{pk}\top} \mid \boldsymbol{F}_a^{\mathsf{sk}\top}\right)^\top$ where $\boldsymbol{F}_a^{\mathsf{pk}} \in \mathbb{F}_q^{l_{\mathsf{pk}} \times l_s}, \boldsymbol{F}_a^{\mathsf{sk}} \in \mathbb{F}_q^{l_{\mathsf{sk}} \times l_s}$. Since we made the assumption that all matrix dimensions are multiples of N, we can decompose $\boldsymbol{F}_a^{\mathsf{pk}}, \boldsymbol{F}_a^{\mathsf{sk}}$ into submatrices of size $N \times N$. Moreover, let $\boldsymbol{r}$ be a vector $\boldsymbol{r} = (\boldsymbol{r}_1 | \ldots | \boldsymbol{r}_k)^\top$ where each $\boldsymbol{r}_i \in R_q$, then $C(\boldsymbol{r})$ is an abbreviation for the list of commitments to each individual $\boldsymbol{r}_i$. This gives an intuitive way to extend $\Pi_{\text{OPEN-X}}, \Pi_{\text{SUM}}$ and Π_{BOUND} to longer vectors. We implicitly assume that if we apply Π_{LIN} to a matrix $\boldsymbol{F}$ and $C(\boldsymbol{r})$, then the appropriate number of individual instances of Π_{LIN} with the respective submatrices of $\boldsymbol{F}$ is being used. Using this generalization, we can instantiate KG with active security as

shown in Fig. 3. We assume that there exists a coin-flipping functionality $\mathcal{F}_{\text{RAND}}$ because the zero-knowledge protocols are only honest-verifier zero-knowledge. The commitments $C(\mathsf{sk}_i) := C(\boldsymbol{F}_a^{\mathsf{sk}} \boldsymbol{s}_i)$ will be saved for later: we will use it again in the distributed decryption.

Formally, one can show the following statement to establish security:

Theorem 1. *The protocol Π_{KG} implements $\mathcal{F}_{\text{KGD}}$ in the standalone setting with security against static active adversaries corrupting up to $n-1$ parties in the $\mathcal{F}_{\text{RAND}}$-hybrid model with auxiliary commitments and broadcast.*

The proof can be found in Appendix D. Moreover, in Appendix E we show how to achieve UC security for this protocol, as well as our decryption protocol.

Acknowledgments

We thank Nigel Smart for informing us about a bug in an earlier version of this work.

References

AJL+12. Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. In *Advances in Cryptology - EUROCRYPT 2012*, pages 483–501, 2012.

Ajt96. Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108. ACM, 1996.

BCK+14. Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *Advances in Cryptology - ASIACRYPT 2014*, pages 551–572, 2014.

BDLN16. Carsten Baum, Ivan Damgård, Kasper Larsen, and Michael Nielsen. How to prove knowledge of small secrets. In *Advances in Cryptology-CRYPTO 2016*. Springer, 2016.

BGV12. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.

BKLP15. Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *Computer Security - ESORICS 2015*, pages 305–325, 2015.

BKP13. Rikke Bendlin, Sara Krehbiel, and Chris Peikert. How to share a lattice trapdoor: Threshold protocols for signatures and (H)IBE. In *Applied Cryptography and Network Security - 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*, pages 218–236, 2013.

Blu82. Manuel Blum. Coin flipping by telephone - A protocol for solving impossible problems. In *COMPCON'82, Digest of Papers, Twenty-Fourth IEEE Computer Society International Conference, San Francisco, California, USA, February 22-25, 1982*, pages 133–137, 1982.

BRS02. John Black, Phillip Rogaway, and Thomas Shrimpton. Encryption-scheme security in the presence of key-dependent messages. In *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*, pages 62–75, 2002.

BV11. Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *Advances in Cryptology–CRYPTO 2011*, pages 505–524. Springer, 2011.

CDXY17. Ronald Cramer, Ivan Damgård, Chaoping Xing, and Chen Yuan. Amortized complexity of zero-knowledge proofs revisited: Achieving linear soundness slack. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, pages 479–500, 2017.

CKLS16. Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yong Soo Song. Lizard: Cut off the tail! // practical post-quantum public-key encryption from LWE and LWR. Cryptology ePrint Archive, Report 2016/1126, 2016. `http://eprint.iacr.org/2016/1126`.

DF89. Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, pages 307–315, 1989.

DKL+13. Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P. Smart. Practical covertly secure MPC for dishonest majority - or: Breaking the SPDZ limits. In *Computer Security - ESORICS 2013 - 18th European Symposium on Research in Computer Security, Egham, UK, September 9-13, 2013. Proceedings*, pages 1–18, 2013.

DPP93. Ivan Damgård, Torben P. Pedersen, and Birgit Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. In *Advances in Cryptology - CRYPTO '93*, pages 250–265, 1993.

DPSZ12. Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty Computation from Somewhat Homomorphic Encryption. In *Proceedings of Crypto*, pages 643–662, Springer Verlag 2012.

FV12. Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206, 2008.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H Silverman. Ntru: A ring-based public key cryptosystem. In *Algorithmic number theory*, pages 267–288. Springer, 1998.

JKPT12. Abhishek Jain, Stephan Krenn, Krzysztof Pietrzak, and Aris Tentes. Commitments and efficient zero-knowledge proofs from learning parity with noise. In *Advances in Cryptology - ASIACRYPT 2012*, pages 663–680, 2012.

KTX08. Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, pages 372–389, 2008.

LM06. Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *Automata, Languages and Programming*, pages 144–155. Springer, 2006.

LN17. Vadim Lyubashevsky and Gregory Neven. One-shot verifiable encryption from lattices. In *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part I*, pages 293–323, 2017.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, pages 1–23, 2010.

LPR13. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. A toolkit for ring-LWE cryptography. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 35–54. Springer, 2013.

LS17. Vadim Lyubashevsky and Gregor Seiler. Partially splitting rings for faster lattice-based zero-knowledge proofs. Cryptology ePrint Archive, Report 2017/523, 2017. `http://eprint.iacr.org/2017/523`.

LSS16. Yehuda Lindell, Nigel P. Smart, and Eduardo Soria-Vazquez. More efficient constant-round multi-party computation from BMR and SHE. In *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, pages 554–581, 2016.

Lyu08. Vadim Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography–PKC 2008*, pages 162–179. Springer, 2008.

Lyu09. Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, pages 598–616, 2009.

Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, pages 738–755, 2012.

Mic02. Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions from worst-case complexity assumptions. In *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, pages 356–365, 2002.

Mic07. Daniele Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007.

PR06. Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *Theory of Cryptography Conference*, pages 145–166. Springer, 2006.

PR07. Chris Peikert and Alon Rosen. *Lattices that admit logarithmic worst-case to average-case connection factors*, pages 478–487. 2007.
PRS17. Chris Peikert, Oded Regev, and Noah Stephens-Davidowitz. Pseudorandomness of ring-lwe for any ring and modulus. *IACR Cryptology ePrint Archive*, 2017.
Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93, 2005.
Was97. Lawrence C Washington. *Introduction to cyclotomic fields*, volume 83. Springer Science & Business Media, 1997.
XXW13. Xiang Xie, Rui Xue, and Minqian Wang. Zero knowledge proofs from ring-lwe. In *Cryptology and Network Security - 12th International Conference, CANS 2013*, pages 57–73, 2013.

A Short and Invertible Elements, continued

In this appendix we will present proofs for the claims in Section 3. Moreover, we present certain parameter sets for which our construction yields large sets E, E'.

A.1 The Commitment-friendly Set

In some cases, for example when R_q is a field, a commitment-friendly set as defined in Definition 3 trivially exists. In order to prove Lemma 1 we will start with a generic upper bound on the size of such a set D:

Remark 3. Let R, R_q be defined as before and $D \subseteq \boldsymbol{B}$ be a commitment-friendly set, then $|D| \leq d$ where $d = \mathrm{ord}_m(q)$.

Proof. Decompose R_q into a product of fields using the decomposition of $\Phi_m(X) = f_1(X) \cdots f_t(X) \mod q$ with $deg(f_i) = d$ and $t = N/d$ (see e.g. [Was97]). Then

$$R_q \simeq \mathbb{F}_q[X]/\langle f_1(X)\rangle \times \cdots \times \mathbb{F}_q[X]/\langle f_t(X)\rangle \simeq (\mathbb{F}_q^d) \times \cdots \times (\mathbb{F}_q^d)$$

For any set $D = \{r_1, \ldots, r_s\}$ we require that

$$\forall a_i \in \mathbb{F}_q : \ \sum_{i=1}^{t} a_i r_i = 0 \ \Leftrightarrow \ a_1 = a_2 = \cdots = a_s = 0.$$

Then $s \leq d$, as each subfield of R_q can be written as a vector space of dimension d - the elements must be independent in each $\mathbb{F}_q[X]/\langle f_i(X)\rangle$ as we can otherwise trivially construct an element that is $0 \mod f_i(X)$. □

To identify the elements from the powerful basis $\boldsymbol{B}$ that we can choose, we need the following, technical statement.

Proposition 3. *Let $m_1, m_2 \in \mathbb{N}$ such that m_2 is a prime power, $m_1 > 1$ and $gcd(m_1, m_2) = 1$. Moreover, let q be a prime, $m = m_1 m_2$ and set*

$$d = \mathrm{ord}_m(q), \ d_1 = \mathrm{ord}_{m_1}(q), \ d_2 = d/d_1$$

Let $\omega_m \in \overline{\mathbb{F}_q}$ be a primitive mth root of unity in the algebraic closure of $\mathbb{F}_q$. Then it must hold for the degree of the extension that

$$[\mathbb{F}_q(\omega_{m_1}, \omega_{m_2}) : \mathbb{F}_q(\omega_{m_1})] = d_2.$$

Proof. The field extension $\mathbb{F}_q(\omega_m)/\mathbb{F}_q$ has degree $d = [\mathbb{F}_q(\omega_m) : \mathbb{F}_q] = \mathrm{ord}_m(q)$ as shown in e.g. [Was97]. As m_1, m_2 are coprime, $\omega_{m_2} \notin \mathbb{F}_q(\omega_{m_1})$. Hence adjoining ω_{m_2} leads to a real field extension. Moreover, $\mathbb{F}_q(\omega_{m_1}, \omega_{m_2}) \subseteq \mathbb{F}_q(\omega_m)$ as we can write both $\omega_{m_1}, \omega_{m_2}$ as powers of ω_m.

The homomorphism

$$\begin{aligned} \phi : \mathbb{F}_q(\omega_{m_1}, \omega_{m_2}) &\to \mathbb{F}_q(\omega_m) \\ a_{i,j}\omega_{m1}^j \omega_{m_2}^i &\mapsto a_{i,j}\omega_m^{d*j/d_1+i} \end{aligned}$$

is injective as $d * j/d_1 + i$ takes all possible values from $\{0, .., d-1\}$ for $j \in \{0, ..., d_1 - 1\}, i \in \{0, .., d_2 - 1\}$, and by the size of $\mathbb{F}_q(\omega_m)$ it must therefore also be surjective. □

This can be used to establish the correctness of Lemma 1:

Proof (of Lemma 1). It is easy to see that $|\mathsf{E}| = d$, which by Remark 3 is maximal. It remains to show that this set indeed commitment-friendly. Consider the tower of extensions

$$\mathbb{F}_q \subset \mathbb{F}_q(\omega_{m_1}) \subset \mathbb{F}(\omega_{m_1}, \omega_{m_2}) \subset \cdots \subset \mathbb{F}_q(\omega_{m_1}, \ldots, \omega_{m_\ell}) \simeq \mathbb{F}_q(\omega_m).$$

As shown in Proposition 3, each of these is a real extension of degree $d_i = \mathrm{ord}_{m_1 \cdots m_i}(q) / \mathrm{ord}_{m_1 \cdots m_{i-1}}(q)$ which means that all powers $\{1, \zeta_{m_i}, ..., \zeta_{m_i}^{d_i - 1}\}$ are linearly independent over $\mathbb{F}(\zeta_{m_1}, ..., \zeta_{m_{i-1}})$. Moreover, we can locate $\mathbb{F}(\omega_m)$ in R_q as it can be decomposed into copies of this field.

Fix the set $\{\zeta_{m_1}, \ldots, \zeta_{m_\ell}\} \subset \overline{\boldsymbol{B}}$ as used in the powerful basis of R_q, and consider the injective homomorphism

$$\begin{aligned} \sigma : \mathbb{F}_q(\omega_{m_1}, \ldots, \omega_{m_\ell}) &\to R_q \\ \sum_{\substack{i \in [\ell] \\ j_i \in \{0,\ldots,d_i-1\}}} a_{j_1,\ldots,j_\ell} \omega_{m_1}^{j_1} \cdots \omega_{m_\ell}^{j_\ell} &\mapsto \sum_{\substack{i \in [\ell] \\ j_i \in \{0,\ldots,d_i-1\}}} a_{j_1,\ldots,j_\ell} \zeta_{m_1}^{j_1} \cdots \zeta_{m_\ell}^{j_\ell} \end{aligned}$$

then the image of σ consists of exactly those $\mathbb{F}_q$-combinations over E. By injectivity, $\sigma(\mathbb{F}_q(\omega_{m_1}, \ldots, \omega_{m_\ell}) \setminus \{0\})$ must be invertible. □

A.2 The Multiplication-friendly Set

In Proposition 1 we gave an explicit construction for a multiplication-friendly set, based on the commitment-friendly set E from Lemma 1. We now give a proof for the claimed statement:

Proof (of Proposition 1). Let $x \in S_{\mathsf{E}',\beta}, z \in R$ such that $||z||_p^c \leq \gamma$. We have that

$$\begin{aligned} ||x \cdot y||_p^c &\leq ||x||_\infty^c \cdot ||y||_p^c \\ &\leq |\mathsf{E}|/2 \cdot ||x||_\infty \cdot ||y||_p^c \\ &= |\mathsf{E}|/2 \cdot \beta \cdot \gamma \end{aligned}$$

because obtaining the canonical embedding consists of computing $\langle x, r \rangle$ where at most $|\mathsf{E}'| \leq |\mathsf{E}|/2$ elements of x are non-zero and $||r||_\infty = 1$.

We obtain each coefficient in the product of the second claim by multiplying at most $|\mathsf{E}'|$ non-zero elements with each other, and thus after the multiplication at most $|\mathsf{E}|$ elements are non-zero, as the result must be in E by construction. □

There are other ways how to obtain a commitment-friendly and multiplication-friendly set: for example, using $m = 2^\tau$ and $q = 3 \mod 8$ as in [BKLP15] one can, considering that the power basis and the powerful basis coincide in this case, choose all polynomials of *small enough* degree and instantiate our construction this way. In this setting R_q decomposes into two large fields. In [LN17] the authors observe that $q = 5 \mod 8$ yields a similar result.

A.3 Properties of the Construction

In a computational setting we assume that $m = O(\lambda)$. It is interesting to discuss the size of E', E in the best and worst case for R_q. Intuitively, one would hope that $|\mathsf{E}'| = \Theta(|\mathsf{E}|)$. We will show that for some interesting cases, E is large and the above intuition holds, while it is not true in general.

Worst case: In the worst case, one can set $q = 1 \bmod m$. Then $|\mathsf{E}| = |\mathsf{E}'| = 1$ as R_q totally decomposes.

Best case: Let $m = p^k$ or $m = 2p^k$ for an odd prime p and $k > 0$. In this setting the group $(\mathbb{Z}/m\mathbb{Z})^*$ is cyclic and therefore contains an element r of order $\varphi(m) = (p-1)p^k$. A prime $q = r \mod m$ must exist due to Dirichlet's Theorem. In this setting we have $|\mathsf{E}'| = |\mathsf{E}| = \Omega(\lambda)$. This is the *other extreme* to the worst case, as R_q is now a field.

Two other interesting cases are if we set $m = 2^k$ or when m is the product of a constant number of powers of safe primes.

Powers of Two. We may ask how big $|\mathsf{E}|$ can be in this setting, so what the largest $\mathrm{ord}_{2^k}(q)$ can be. The value that can be achieved is called *the Carmichael Number*[10] $\chi(\cdot)$, which is the smallest value $\chi(n)$ such that $\forall q \in \mathbb{N}^+ : \ gcd(q, n) = 1 \Rightarrow q^{\chi(n) \bmod n}$. For powers of two one has $\chi(2^k) = \varphi(2^k)/2 = 2^{k-2}$. We can therefore assume the existence of a prime q that has order $2^{k-2} = O(\lambda)$ and obtain that also $|\mathsf{E}| = |\mathsf{E}'| = \Omega(\lambda)$.

[10] In the literature one denotes the Carmichael Number of n usually using $\lambda(n)$.

Products of Safe Primes. Let $m = m_1^{k_1} \cdots m_\ell^{k_\ell}$ with $t = O(1)$. Assume that $\forall i \in [t] : \ \frac{m_i - 1}{2} \not| \ m$. Moreover $\chi(m) = lcm(\chi(m_1^{k_1}), \ldots, \chi(m_\ell^{k_\ell}))$. But then

$$\chi(m) = 2p_1 m_1^{k_1 - 1} \cdots p_\ell m_\ell^{k_\ell} = |\mathsf{E}| = O(\lambda)$$

Let $m_1, \ldots, m_\ell$ be in ascending order, then m_1 would be the smallest safe prime. One can easily check that in this case $\lfloor \frac{d_i - 1}{2} \rfloor \geq \frac{d_i}{4}$. Since $t = O(1)$ we yield $|\mathsf{E}'| \geq |\mathsf{E}|/4^\ell = \Omega(\lambda)$.

B Extending the Commitment Scheme to Vectors

It was briefly mentioned in Section 4 that the commitment scheme can be extended to allow committing to vectors $\boldsymbol{x} \in R_q^d$ (for constant d). We will describe the modification in this appendix.

CKeyGen: Output the specification of a ring R_q with the commitment-friendly set D from Definition 3 and a (s,t)-multiplication-friendly set D' as defined in Definition 4, a uniformly random matrix $\boldsymbol{A} \in R_q^{(d+1)\times k}$ and a constant β such that $16 \cdot s \cdot t \cdot k \cdot m \cdot \log(k \cdot N) \cdot \beta^3 < q/2$ Finally, we define the distribution $\mathcal{D}$ which outputs a vector $\boldsymbol{v} \in S^k_{D',\beta}$ uniformly at random.

Commit: To commit to a message $x \in R_q^d$, draw a $\boldsymbol{r} \xleftarrow{\$} \mathcal{D}$ and compute

$$Com(x; \boldsymbol{r}) := \boldsymbol{A}\boldsymbol{r} + \binom{0}{x}.$$

Open: A valid opening of a commitment $\boldsymbol{c}$ is a 3-tuple: $x \in R_q^d$, $\boldsymbol{r} \in R^k$, and $\mathsf{f} \in S_{D',2\cdot\beta}$. The verifier checks that

$$\boldsymbol{A}\boldsymbol{r} + \binom{0}{\mathsf{f}x} = \mathsf{f}\boldsymbol{c},$$

and that $||\boldsymbol{r}||_2^c \leq 4 \cdot t \cdot k \cdot m \cdot \log(k \cdot N) \cdot \beta^2$.

The security properties of the extended commitment scheme remain similar:

Lemma 5 (Binding Property). *From a commitment $\boldsymbol{c}$ and correct openings $\boldsymbol{r}, \mathsf{f}, \boldsymbol{r}', \mathsf{f}'$ to two different messages x, x', one can efficiently compute a solution with $16 \cdot s \cdot t \cdot k \cdot m \cdot \log(k \cdot N) \cdot \beta^3$ to the Ring-SIS problem instance defined by the top row of $\boldsymbol{A}$.*

Lemma 6 (Hiding Property). *Assume the distribution $\mathcal{D}$ and that R_q, k are chosen such that 1) the min-entropy of a vector drawn from $\mathcal{D}$ is at least $(d+1) \cdot \log(|R_q|) + \kappa$ where κ is a (statistical) security parameter, and 2) the class of functions $\{f_{\boldsymbol{a}} \mid \boldsymbol{a} \in R_q^k\}$ where $f_{\boldsymbol{a}}(\boldsymbol{r}) = \boldsymbol{a} \cdot \boldsymbol{r}$ is universal when mapping the support of $\mathcal{D}$ to R_q. Then the scheme is statistically hiding.*

The proofs of the binding and hiding property follow by the same arguments as for Lemma 2 and Lemma 3.

C Proofs for Zero-Knowledge Proofs

In this appendix we will provide a proof for Lemma 4, as well as protocols and proofs for other applications as outlined in Section 5. To ease understanding of these proofs, we first repeat two core results from [Lyu12] that will be useful in our context.

We adapt the tail-bound from [Lyu12, Lemma 4.4] as

Remark 4. For any $K > 0$,

$$\Pr[||\boldsymbol{z}||_2 > K\sigma\sqrt{kN} \mid \boldsymbol{z} \xleftarrow{\$} \mathcal{N}_\sigma^k] < K^{kN} \cdot \exp\left(\frac{kN}{2}(1-K^2)\right).$$

Moreover, the rejection sampling theorem from [Lyu12, Theorem 4.6] can be expressed in our setting as follows:

Lemma 7. *Let $V \subseteq R^k$ such that all elements have $||\cdot||_2$-norm less than T, $\sigma \in \mathbb{R}$ such that $\sigma = \omega(T\sqrt{\log(kN)})$ and $h : V \to \mathbb{R}$ be a probability distribution. Then there exists a $M = O(1)$ such that the distribution of the following two algorithms $\mathcal{A}, \mathcal{S}$ is within statistical distance $2^{-\omega(\log(kN))}/M$.*

$\mathcal{A}$**:**

1. $\boldsymbol{v} \xleftarrow{\$} h$
2. $\boldsymbol{z} \xleftarrow{\$} \mathcal{N}_{\boldsymbol{v},\sigma}^k$
3. *Output* (z, v) *with probability* $\min\left(\frac{\mathcal{N}_\sigma^k(\boldsymbol{z})}{M\mathcal{N}_{\boldsymbol{v},\sigma}^k(\boldsymbol{z})}, 1\right)$

$\mathcal{S}$**:**

1. $\boldsymbol{v} \xleftarrow{\$} h$
2. $\boldsymbol{z} \xleftarrow{\$} \mathcal{N}_\sigma^k$
3. *Output* (z, v) *with prob.* $1/M$

The probability that $\mathcal{A}$ outputs something is at least $\dfrac{1 - 2^{\omega(\log(kN))}}{M}$.

As mentioned in [Lyu12], by setting $\sigma = \alpha T$ one obtains

$$M = \exp\left(12/\alpha + 1/(2\alpha^2)\right)$$

such that the statistical distance of the output of $\mathcal{A}, \mathcal{S}$ is at most $2^{-100}/M$ while $\mathcal{A}$ outputs a result with probability at least $(1-2^{-100})/M$. In practice one would choose $kN \gg 128$, but already for $kN = 128$ one obtains that $M \approx 4.5$, and it just decreases for larger choices.

In Π_{OPEN} as well as the other protocols, we set $K = 2$. This choice is sufficient for Remark 4 as we surely have $N = \Omega(\lambda)$, so the tail-bound holds with probability that is overwhelming in λ.

C.1 Proof for Opening a Commitment

Proof (of Lemma 4).

Completeness: An honest prover can clearly answer correctly for any challenge d and by Lemma 7, the abort probability of the sender for our choice of parameters is at most $1 - \frac{1-2^{-100}}{M}$. For the receiver, by Remark 4 the bound on $||\boldsymbol{r}_z||_2$ is $2\sigma \cdot \sqrt{kN}$ except with negligible probability.

Special Soundness: We first note that if $t \neq t'$ in the input information, this breaks binding for the auxiliary scheme (of course one expects that this occurs with negligible probability). Otherwise $t = t'$, and one can compute the message contained in $\boldsymbol{c}$ as $x = \mathsf{f}^{-1}(z - z')$ where $\mathsf{f} = \mathsf{d} - \mathsf{d}'$ and is indeed invertible as $\mathsf{f} \in S_{D',\beta} \setminus \{0\}$. We set the randomness to be $\boldsymbol{r} = \boldsymbol{r}_z - \boldsymbol{r}'_z$.

This works since if we subtract the two equations the verifier would check in the two transcripts, we obtain

$$(\mathsf{d} - \mathsf{d}')\boldsymbol{c} = A(\boldsymbol{r}_z - \boldsymbol{r}'_z) + \left(\begin{smallmatrix} 0 \\ z - z' \end{smallmatrix}\right),$$

which by definition of f and $\boldsymbol{r}_\mu$ can be rewritten to

$$\mathsf{f}\boldsymbol{c} = A\boldsymbol{r} + \left(\begin{smallmatrix} 0 \\ \mathsf{f}x \end{smallmatrix}\right).$$

So the opening information we obtain is $x, \mathsf{f}, \boldsymbol{r}$.

Honest-Verifier Zero-Knowledge: The simulator first decides to simulate an aborting conversation with probability $1/M$. In this case, the simulator just outputs $C_{aux}(t)$ for an arbitrary value t of the same length as a basic commitment.

Otherwise, to simulate an accepting conversation, draw a random d from $S_{D',\beta}$ and a random $\boldsymbol{r}_z$ from $\mathcal{N}_\sigma^k$. Finally, set $\boldsymbol{t} = C(z; \boldsymbol{r}_z) - \mathsf{d}\boldsymbol{c}$, and commit to $\boldsymbol{t}$ using the auxiliary commitment scheme. As for correctness of the output distribution, note that aborting and non-aborting conversations occur with the correct probability. The aborting conversations have statistically indistinguishable distribution by the statistical hiding of the auxiliary commitment scheme. The non-aborting ones are statistically indistinguishable as the simulator simply acts as $\mathcal{S}$ as in Lemma 7. □

C.2 Proof of Linear Relation

Protocol $\boldsymbol{\Pi}_{\mathbf{Lin}}$

1. The prover computes commitments $\boldsymbol{t}_1 = C(\mu_1; \boldsymbol{\rho}_1), \boldsymbol{t}_2 = C(\mu_2; \boldsymbol{\rho}_2)$ where μ_1 is chosen uniformly from R_q, and $\boldsymbol{\rho}_1, \boldsymbol{\rho}_2$ are sampled from $\mathcal{N}_\sigma^k$. Furthermore, set $\mu_2 = g(\mu_1)$. The prover then sends the commitments $\boldsymbol{c}_{aux,1} = C_{aux}(\boldsymbol{t}_1)$, $\boldsymbol{c}_{aux,2} = C_{aux}(\boldsymbol{t}_2)$ to the verifier.
2. The verifier sends a random challenge $\mathsf{d} \in S_{D',\beta}$.
3. The prover first checks that d is a valid challenge. The prover's goal is to open $\boldsymbol{t}_1 + \mathsf{d}\boldsymbol{c}_1$ to $z_1 = \mu_1 + \mathsf{d}x_1$ and $\boldsymbol{r}_{z_1} = \boldsymbol{\rho}_1 + \mathsf{d}\boldsymbol{r_1}$, and $\boldsymbol{t}_2 + \mathsf{d}\boldsymbol{c}_2$ to $z_2 = \mu_2 + \mathsf{d}x_2$ and $\boldsymbol{r}_{z_2} = \boldsymbol{\rho}_2 + \mathsf{d}\boldsymbol{r}_2$. The protocol is aborted with probability

$$1 - \min\left(1, \frac{\mathcal{N}_\sigma^k(\boldsymbol{r}_{z_1})}{M\mathcal{N}_{\mathsf{d}\boldsymbol{r}_1,\sigma}^k(\boldsymbol{r}_{z_1})} + \frac{\mathcal{N}_\sigma^k(\boldsymbol{r}_{z_2})}{M\mathcal{N}_{\mathsf{d}\boldsymbol{r}_2,\sigma}^k(\boldsymbol{r}_{z_2})}\right).$$

Otherwise, the prover sends to the verifier z_1, $\boldsymbol{r}_{z_1}$, z_2, $\boldsymbol{r}_{z_2}$, and opening information $u_{aux,1}$ and $u_{aux,2}$ for the commitments $\boldsymbol{c}_{aux,1}$ and $\boldsymbol{c}_{aux,2}$.

4. The verifier checks that $u_{aux,1}$, $u_{aux,2}$ are valid, that $C(z_1; \boldsymbol{r}_{z_1}) = \boldsymbol{t}_1 + \mathsf{d}\boldsymbol{c}_1$ and $C(z_2; \boldsymbol{r}_{z_2}) = \boldsymbol{t}_2 + \mathsf{d}\boldsymbol{c}_2$, that $g(z_1) = z_2$, and that $||\boldsymbol{r}_{z_1}||_2, ||\boldsymbol{r}_{z_2}||_2 \leq 2\sigma\sqrt{kN}$.

Lemma 8. *The protocol Π_{LIN} has the following properties:*

- Completeness: *The verifier accepts an interaction with an honest prover with overwhelming probability when the protocol does not abort. The probability of abort is at most* $1 - 2\frac{1-2^{-100}}{M}$.
- Special Soundness: *On input two commitments $\boldsymbol{c}_1$, $\boldsymbol{c}_2$ and a pair of transcripts* $((\boldsymbol{c}_{aux,1}, \boldsymbol{c}_{aux,2}), \mathsf{d}, (u_{aux,1}, u_{aux,2}, \boldsymbol{t}_1, \boldsymbol{t}_2, z_1, z_2, \boldsymbol{r}_{z_1}, \boldsymbol{r}_{z_2}))$, $((\boldsymbol{c}_{aux,1}, \boldsymbol{c}_{aux,2}), \mathsf{d}', (u'_{aux,1}, u'_{aux,2}, \boldsymbol{t}'_1, \boldsymbol{t}'_2, z'_1, z'_2, \boldsymbol{r}'_{z_1}, \boldsymbol{r}'_{z_2}))$ *where $\mathsf{d} \neq \mathsf{d}'$, we can extract either a witness for breaking the auxiliary commitment scheme, or valid openings of $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$.*
- Honest-Verifier Zero-Knowledge: *Executions of protocol Π_{LIN} with an honest verifier can be simulated with statistically indistinguishable distribution.*

Proof. An honest prover can clearly answer with both values correctly for any challenge d and by Lemma 7, the abort probability of the sender for our choice of parameters is at most $1 - 2\frac{1-2^{-100}}{M}$ as both values are computed independently. For the receiver, by Remark 4 one can again give a bound on the $||\cdot||_2$-norms on the returned values $\boldsymbol{r}_{z_1}, \boldsymbol{r}_{z_2}$ as in the proof of Lemma 4.

The proof of special soundness is similar to that of Lemma 4: if we cannot break the auxiliary commitment scheme, then by the same argument, we can assume that $\boldsymbol{t}_1 = \boldsymbol{t}'_1$ and $\boldsymbol{t}_2 = \boldsymbol{t}'_2$. In this case, one can compute the messages contained in $\boldsymbol{c}_1$ as $x_1 = \mathsf{f}^{-1}z_1 - z'_1$ and in $\boldsymbol{c}_2$ as $x_2 = \mathsf{f}^{-1}z_2 - z'_2$, where $\mathsf{f} = \mathsf{d} - \mathsf{d}'$ and f is again invertible. Then set the randomness $\boldsymbol{r}_1 = \boldsymbol{r}_{z_1} - \boldsymbol{r}'_{z_2}$ and $\boldsymbol{r}_2 = \boldsymbol{r}_{z_2} - \boldsymbol{r}'_{z_2}$.

For honest-verifier zero-knowledge, note that the probability p_{abort} that an abort occurs in the protocol is independent of the prover's secret. Therefore, on input $\boldsymbol{c}_1, \boldsymbol{c}_2$, the simulator first decides to simulate an aborting conversation with probability p_{abort}. In this case, the simulator just outputs $C_{aux}(s)$ and $C_{aux}(t)$ for arbitrary values s and t of the same length as a basic commitment.

Otherwise, to simulate an accepting conversation, draw a random d from $S_{D',\beta}$, a random z_1, and $\boldsymbol{r}_{z_1}$, $\boldsymbol{r}_{z_2}$ from $\mathcal{N}_\sigma^k$. Set $z_2 = g(z_1)$. Finally, set $\boldsymbol{t}_1 = C(z_1; \boldsymbol{r}_{z_1}) - \mathsf{d}\boldsymbol{c}_1$ and $\boldsymbol{t}_2 = C(z_2; \boldsymbol{r}_{z_2}) - \mathsf{d}\boldsymbol{c}_2$, and commit to $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ using the auxiliary commitment scheme. As for correctness of output distribution, note that aborting and non-aborting conversations occur with the correct probabilities. The aborting conversations have statistically indistinguishable distribution by hiding of the auxiliary scheme. Indistinguishability of the non-aborting conversations follows by the same argument as in the proof of Lemma 4. □

C.3 Proof of Sum

Protocol Π_{Sum}

1. The prover draws uniform μ_1, μ_2 from R_q and $\boldsymbol{\rho}_i$ ($i \in \{1,2,3\}$) from $\mathcal{N}_\sigma^k$, and sets $\mu_3 = \alpha_1\mu_1 + \alpha_2\mu_2$. He then computes $\boldsymbol{t}_i = C(\mu_i; \boldsymbol{\rho}_i)$ and $\boldsymbol{c}_{aux,i} = C_{aux}(\boldsymbol{t}_i)$ ($i \in \{1,2,3\}$). Finally, the prover sends $\boldsymbol{c}_{aux,i}$ to the verifier.
2. The verifier sends a random challenge $\mathsf{d} \in S_{D',\beta}$.
3. The prover first checks that d is a valid challenge. The prover's goal is then to open $\boldsymbol{t}_i + \mathsf{d}\boldsymbol{c}_i$ to $z_i = \mu_i + \mathsf{d}x_i$ and $\boldsymbol{r}_{z_i} = \boldsymbol{\rho}_i + \mathsf{d}\boldsymbol{r}_i$. The protocol is aborted with probability

$$1 - \min\left(1, \frac{\mathcal{N}_\sigma^k(\boldsymbol{r}_{z_1})}{M\mathcal{N}_{\mathsf{d}r,\sigma}^k(\boldsymbol{r}_{z_1})} + \frac{\mathcal{N}_\sigma^k(\boldsymbol{r}_{z_2})}{M\mathcal{N}_{\mathsf{d}r,\sigma}^k(\boldsymbol{r}_{z_2})} + \frac{\mathcal{N}_\sigma^k(\boldsymbol{r}_{z_3})}{M\mathcal{N}_{\mathsf{d}r,\sigma}^k(\boldsymbol{r}_{z_3})}\right).$$

   Otherwise, the prover sends to the verifier z_i, $\boldsymbol{r}_{z_i}$, and opening information $u_{aux,i}$ for the commitments $\boldsymbol{c}_{aux,i}$.
4. The verifier checks that $u_{aux,i}$ are valid, that $C(z_i; \boldsymbol{r}_{z_i}) = \boldsymbol{t}_i + \mathsf{d}\boldsymbol{c}_i$, that $z_3 = \alpha_1 z_1 + \alpha_2 z_2$, and that $||\boldsymbol{r}_{z_i}||_2 \leq 2\sigma\sqrt{kN}$ for $i \in \{1,2,3\}$.

Lemma 9. *The protocol Π_{SUM} has the following properties:*

- Correctness: *The verifier accepts an interaction with an honest prover with overwhelming probability when the protocol does not abort. The probability of abort is at most $1 - 3\frac{1-2^{-100}}{M}$.*
- Special Soundness: *On input α_1, α_2, three commitments $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$ and a pair of transcripts $((\boldsymbol{c}_{aux,i})_{i\in\{1,2,3\}}, \mathsf{d}, (u_{aux,i}, \boldsymbol{t}_i, z_i, \boldsymbol{r}_{z_i})_{i\in\{1,2,3\}})$, $((\boldsymbol{c}_{aux,i})_{i\in\{1,2,3\}}, \mathsf{d}', (u'_{aux,i}, \boldsymbol{t}'_i, z'_i, \boldsymbol{r}'_{z_i})_{i\in\{1,2,3\}})$ where $\mathsf{d} \neq \mathsf{d}'$, we can extract either a witness for breaking the auxiliary commitment scheme, or valid openings of $\boldsymbol{c}_1$, $\boldsymbol{c}_2$, $\boldsymbol{c}_3$.*
- Honest-Verifier Zero-Knowledge: *Executions of protocol Π_{SUM} with an honest verifier can be simulated with statistically indistinguishable distribution.*

Proof. As argued above, the output of the honest prover for every challenge will be accepted by an honest verifier except with probability negligible in λ, while the abort probability is at most $1 - 3\frac{1-2^{-100}}{M}$.

The proof of special soundness is similar to that of Lemma 4: if we cannot break the auxiliary commitment scheme, then by the same argument, we can assume that $\boldsymbol{t}_i = \boldsymbol{t}'_i$. In this case, one can compute the messages contained in $\boldsymbol{c}_i$ as $x_i = \mathsf{f}^{-1}(z_i - z'_i)$, where $\mathsf{f} = \mathsf{d} - \mathsf{d}'$ and f is again invertible. Then set the randomness $\boldsymbol{r}_i = \boldsymbol{r}_{z_i} - \boldsymbol{r}'_{z_i}$.

For honest-verifier zero-knowledge, first note that the probability p_{abort} that an abort occurs in the protocol is independent of the prover's secret (cf. Lemma 4). Therefore, on input $\boldsymbol{c}_i$, the simulator first decides to simulate an aborting conversation with probability p_{abort}. In this case, the simulator just outputs $C_{aux}(\boldsymbol{t}_i)$ for arbitrary values $\boldsymbol{t}_i$ of the same length as a basic commitment.

Otherwise, to simulate an accepting conversation, draw a random d from $S_{D',\beta}$ and random z_1, z_2, and $\boldsymbol{r}_{z_i}$ from $\mathcal{N}_\sigma^k$. Set $z_3 = \alpha_1 z_1 + \alpha_2 z_2$. Finally, set $\boldsymbol{t}_i = C(z_i; \boldsymbol{r}_{z_i}) - \mathsf{d}\boldsymbol{c}_i$, and commit to $\boldsymbol{t}_i$ using the auxiliary commitment scheme. As for correctness of output distribution, note that aborting and non-aborting conversations occur with the correct probabilities. The aborting conversations

have statistically indistinguishable distribution by hiding of the auxiliary scheme. Indistinguishability of the non-aborting conversations follows by the same argument as in the proof of Lemma 4. □

C.4 Proving Bounds

Let β_x be an upper bound on the norms of all possible x and β_r an upper bound on the norm of the possible μ, where $\beta_r \geq \gamma_x N \beta_x$ for $\gamma_x > 0$. In addition to the rejection sampling to hide the randomness of the commitment, we now perform an additional rejection sampling which hides the actual message and just proves its size. But the bound on the message is defined in the ℓ_∞-norm and we use a technique similar to [Lyu08,Lyu09].

Protocol Π_{Bound}

1. The prover computes a commitment $\boldsymbol{t} = C(\mu; \boldsymbol{\rho})$ for uniform $\mu \in R$ and $\boldsymbol{\rho} \in R^k$ from $\mathcal{N}_\sigma^k$, subject to $||\mu||_\infty \leq \beta_x(1 + \gamma_x N/2)$, and sends $c_{aux} = C_{aux}(\boldsymbol{t})$ to the verifier.
2. The verifier sends a random challenge bit $\mathsf{d} \in \{0, 1\}$.
3. The prover first checks that d is a valid challenge. The prover's goal is then to open $\boldsymbol{t} + \mathsf{d}\boldsymbol{c}$ to $z = \mu + dx$ and $\boldsymbol{z}_r = \boldsymbol{\rho} + \mathsf{d}\boldsymbol{r}$. The protocol is aborted with probability $1 - \min(1, \frac{\mathcal{N}_\sigma^k(\boldsymbol{r}_z)}{M\mathcal{N}_{\mathsf{d}r,\sigma}^k(\boldsymbol{r}_z)})$ or if $||z||_\infty > \gamma_x N \beta_x/2$. Otherwise, the prover sends to the verifier z, $\boldsymbol{r}_z$, and opening information u_{aux} for the commitment $\boldsymbol{c}$.
4. The verifier checks that u_{aux} is valid, that $C(z; \boldsymbol{r}_z) = \boldsymbol{t} + \mathsf{d}\boldsymbol{c}$, that $||z||_\infty \leq (\gamma_x N \beta_x/2)$, and that $||\boldsymbol{r}_z||_2 \leq 2\sigma\sqrt{kN}$.

Lemma 10. *The protocol Π_{Bound} has the following properties:*

- Correctness: *The verifier accepts an interaction with an honest prover with overwhelming probability when the protocol does not abort. The probability of abort is at most* $1 - \frac{1-2^{-100}}{M} + 2/\gamma_x$.
- Special soundness: *On input commitment $\boldsymbol{c}$ and a pair of transcripts $(\boldsymbol{c}_{aux}, \mathsf{d}, (u_{aux}, \boldsymbol{t}, z, \boldsymbol{r}_z))$, $(\boldsymbol{c}_{aux}, \mathsf{d}', (u'_{aux}, \boldsymbol{t}', z', \boldsymbol{r}'_z))$ where $\mathsf{d} \neq \mathsf{d}'$, we can extract either a witness for breaking the auxiliary commitment scheme, or a valid opening of $\boldsymbol{c}$ where the message x has norm at most $\gamma_x N \beta_x$.*
- Honest-verifier zero-knowledge: *Executions of protocol Π_{Bound} with an honest verifier can be simulated with statistically indistinguishable distribution.*

Proof. As argued above, the output of the honest prover for every challenge will be accepted by an honest verifier except with probability negligible in λ, while the abort probability due to the Gaussian sampling is at most $1 - \frac{1-2^{-100}}{M}$. For the probability of aborting because the norm of the committed message is too big, note that the challenge d is a bit in this case and hence has norm at most 1. The probability that a single coefficient of $\boldsymbol{z}$ will cause an abort is

$$\frac{2\beta_x}{2(1 + \gamma_x N/2)\beta_x + 1} \leq \frac{2}{\gamma_x N}$$

since $z \in R_q$ and each coefficient of z has norm at most $\gamma_x N\beta_x$. Hence by the union bound, the probability that some coefficient of z or $\boldsymbol{r}_z$ causes an abort is at most $2/\gamma_x$. This yields an overall abort probability $p_{abort} = 1 - \frac{1-2^{-100}}{M} + 2/\gamma_x$.

The proof of special soundness is similar to that of Lemma 4: if we cannot break the auxiliary commitment scheme, then by the same argument, we can assume that $\boldsymbol{t} = \boldsymbol{t}'$. In this case, one can compute the message contained in $\boldsymbol{c}$ as $x = z - z'$. Then set the randomness $\boldsymbol{r} = \boldsymbol{r}_z - \boldsymbol{r}'_z$. Note that indeed, $||x||_\infty \leq \gamma_x N\beta_x$, $||\boldsymbol{r}||_\infty \leq 2 \cdot \sigma \cdot \sqrt{kN}$ as required.

For honest-verifier zero-knowledge, note that the probability p_{abort} that an abort occurs in the protocol is independent of the prover's secret. Therefore, on input $\boldsymbol{c}$, the simulator first decides to simulate an aborting conversation with probability p_{abort}. In this case, the simulator just outputs $C_{aux}(\boldsymbol{t})$ for an arbitrary value $\boldsymbol{t}$ of the same length as a basic commitment and then aborts.

Otherwise, to simulate an accepting conversation, draw a random d from $\{0, 1\}$ as well as random z and $\boldsymbol{r}_z$ from $\mathcal{N}_\sigma^k$ subject to $||z||_\infty \leq \gamma_x N\beta_x/2$. Finally, set $\boldsymbol{t} = C(z; \boldsymbol{r}_z) - \mathsf{d}\boldsymbol{c}$ and commit to $\boldsymbol{t}$ using the auxiliary commitment scheme. By the same argument as before, we obtain the correct distributions of z, $\boldsymbol{r}_z$ except with negligible probability (if the protocol aborts), and we stop the protocol with the right probability since the events that the protocol would abort to hide the values of z, $\boldsymbol{r}_z$ are independent, therefore the union bound yields the correct probability. □

C.5 Achieving Zero-Knowledge for Dishonest Verifiers

One easy way to have our protocols be zero-knowledge against dishonest verifiers is if a trusted source of random bits is available (which can be implemented via a coin-flipping protocol). One gets the challenge from this source and then clearly honest-verifier zero-knowledge is sufficient.

A different approach is possible if a trapdoor commitment scheme C_{trap} is available, where commitments in this scheme can be equivocated if the trapdoor is known. Then we can transform each of our protocols to a new one that is zero-knowledge: the prover commits to the first message a using C_{trap}, gets the challenge d, then opens $C_{trap}(a)$ and answers d. If the simulator knows the trapdoor, it can make a fake commitment first. Once d arrives, it runs the simulation and equivocates the initial commitment to the value of a that it wants.

D More Distributed Key Generation and Decryption

In this appendix, we extend the results from Section 6: the security of the distributed key generation will be proven and a protocol for actively secure threshold decryption will be presented. To ease readability, we present a short list of all parameters and their meaning in Fig. 4.

Parameter	Explanation
P_i	Party i
$\mathcal{P}$	Set of parties
$\mathcal{I}$	Set of corrupted parties
n	Number of parties
$\mathcal{A}$	Adversary
d	Dimension of the plaintext space $\{0,1\}^d$ of the cryptosystem
l_s	Length of the randomness that goes into key generation
l_c	Length of a ciphertext
$l_{\mathsf{pk}}, l_{\mathsf{sk}}$	Length of the public and private key
l_r	Length of the noise vector that protects the decryption key
l_d	Length of the decryption, before being decoded into a plaintext
β_s, β_d	Maximal norm of randomness used in key generation and the noise used in distributed decryption
ω_s, ω_d	"Slack" in norm between honestly chosen vectors and guarantees of Π_{BOUND} in key generation and decryption
$\mathcal{U}_\beta^\ell$	Uniform distribution for vectors of length ℓ and norm at most β
$\boldsymbol{F}_a^{\mathsf{KG}}, \boldsymbol{F}_r, \boldsymbol{F}_c$	Matrices applied in key generation and decryption

Fig. 4. Parameters and Notation used in this Appendix.

D.1 Threshold Decryption

An actively secure version of Dec can be obtained using a similar compilation step to the one that turned KG into Π_{KG}. The main difference lies in the computed values and in the zero-knowledge proofs that are applied. We moreover use the commitments $C(\mathsf{sk}_i)$ that were generated in Π_{KG} and are publicly known. This allows each party to prove that it has applied $\boldsymbol{F}_c$ to the correct key share. The protocol can be found in Fig. 5.

D.2 Proof of Security of Π_{KG}, Π_{Dec}

We now prove security for both protocols.

Theorem 2. *The protocols Π_{KG}, Π_{DEC} implement $\mathcal{F}_{\text{KGD}}$ in the standalone setting with security against static active adversaries corrupting up to $n-1$ parties in the $\mathcal{F}_{\text{RAND}}$-hybrid model with auxiliary commitments and broadcast.*

A simulator for the protocols is provided in Fig. 6. In the proof, we will argue about the indistinguishability of certain distributions, where $\overset{c}{\approx}$ symbolizes that two distributions are *computationally indistinguishable*. Similarly, we use $\overset{s}{\approx}, \overset{p}{\approx}$ if the distributions are statistically close or perfectly indistinguishable.

Protocol $\Pi_{\mathbf{Dec}}$

The parties in $\mathcal{P}$ want to decrypt the ciphertext c. P_i has sk_i. The commitment $C(\mathsf{sk}_i)$ is known to each $P_j \in \mathcal{P}$.

1. Each P_i locally samples $\boldsymbol{e}_i \xleftarrow{\$} \mathcal{U}^{l_r}_{\beta_d}$.
2. Each P_i derives $\boldsymbol{F}_c$ from c and computes and broadcasts the commitments
$$C(\boldsymbol{F}_c\mathsf{sk}_i), C(\boldsymbol{e}_i), C(\boldsymbol{F}_r\boldsymbol{e}_i), C(\boldsymbol{F}_c\mathsf{sk}_i + \boldsymbol{F}_r\boldsymbol{e}_i).$$
3. Each P_i uses the following proofs towards all parties. Sample the challenge using $\mathcal{F}_{\text{RAND}}$:
   (a) Prove using Π_{BOUND} about $C(\boldsymbol{e}_i)$ that $||\boldsymbol{e}_i||_\infty \leq \beta_d$.
   (b) Prove using Π_{LIN} that $C(\boldsymbol{F}_c\mathsf{sk}_i)$ is the linear transform of $C(\mathsf{sk}_i)$ when applying $\boldsymbol{F}_c$ and that $C(\boldsymbol{F}_r\boldsymbol{e}_i)$ can be obtained from $C(\boldsymbol{e}_i)$ using $\boldsymbol{F}_r$.
   (c) Prove using Π_{SUM} that $C(\boldsymbol{F}_c\mathsf{sk}_i + \boldsymbol{F}_r\boldsymbol{e}_i)$ is the sum of $C(\boldsymbol{F}_c\mathsf{sk}_i)$ and $C(\boldsymbol{F}_r\boldsymbol{e}_i)$.
   If one of the proofs fails then abort.
4. Each P_i broadcasts $\boldsymbol{d}_i$ and proves towards all parties that $C(\boldsymbol{F}_c\mathsf{sk}_i + \boldsymbol{F}_r\boldsymbol{e}_i)$ opens as $\boldsymbol{d}_i$ using $\Pi_{\text{OPEN-X}}$.
5. If all proofs were correct then output $m \leftarrow \mathsf{decode}(\sum_{i\in[n]} \boldsymbol{d}_i)$.

Fig. 5. Π_{DEC}: Protocol for the actively secure decryption of a ciphertext.

Proof (of Theorem 2). We will first prove security of Π_{KG} by showing that the distribution τ_Π of protocol transcripts of Π_{KG} is indistinguishable from the distribution τ_{SIM} of outputs of $\mathcal{S}_{\text{KGD}}$ using a sequence of hybrids.

Key generation. We start by defining $\tau_{1,\text{KG}}$ to be the transcript of a simulator that is the same as τ_{SIM} except that it now aborts if the proofs in Steps $(3)-(4)$ of Π_{KG} are not correct, i.e. we do not specifically check the relations on the extracted data anymore. Because $\Pi_{\text{LIN}}, \Pi_{\text{OPEN-X}}$ are computationally sound we get that $\tau_{\text{SIM}} \overset{c}{\approx} \tau_{1,\text{KG}}$. Define $\tau_{2,\text{KG}}$ to be the same as the simulator for $\tau_{1,\text{KG}}$ except that we now simulate the proofs in Step (3) of Π_{KG}. Because $\Pi_{\text{BOUND}}, \Pi_{\text{LIN}}$ are statistical zero-knowledge, it follows that $\tau_{1,\text{KG}} \overset{s}{\approx} \tau_{2,\text{KG}}$.

Now we start from the other side: define $\tau'_{1,\text{KG}}$ to be the same as τ_Π except that we replace the honest proofs in Steps $(3)-(4)$ with simulations. Therefore $\tau_\Pi \overset{s}{\approx} \tau'_{1,\text{KG}}$. Because we do now not need witnesses anymore, we can define $\tau'_{2,\text{KG}}$ to be the same as $\tau'_{1,\text{KG}}$ except that the commitments in Step (2) are replaced with those generated by $\mathcal{S}_{\text{KGD}}$. By the statistical hiding of the commitment scheme, it holds that $\tau'_{1,\text{KG}} \overset{s}{\approx} \tau'_{2,\text{KG}}$. Moreover, the distributions of $\tau_{2,\text{KG}}$ and $\tau'_{2,\text{KG}}$ are identical and the claim follows.

Decryption. We start similarly as in the Π_{KG} case: first, let $\tau_{1,\text{DEC}}$ be a simulator that does the same as $\mathcal{S}_{\text{KGD}}$, but aborts in Steps $(3)-(4)$ only if one of the proofs aborts. We obtain hat $\tau_{\text{SIM}} \overset{c}{\approx} \tau_{1,\text{DEC}}$ due to the computational binding property. In particular, this means that in $\tau_{1,\text{DEC}}$ the adversary must use the

Simulator $\mathcal{S}_{\mathbf{KGD}}$

Key Generation:

1. Wait for $\mathcal{A}$ to input the set $\mathcal{I}$ of corrupted parties.
2. For each honest $P_i \in \mathcal{P} \setminus \mathcal{I}$ choose $\boldsymbol{s}_i \leftarrow \mathcal{U}_{\beta_s}^{l_s}$.
3. For each honest P_i compute the commitments $C(\boldsymbol{s}_i), C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_i), C(\boldsymbol{F}_a^{\mathsf{sk}}\boldsymbol{s}_i)$ and send them to all dishonest parties P_j.
4. For each honest party P_i perform the zero-knowledge proofs in Step (3) of Π_{KG} honestly. Abort if the protocol aborts.
5. Rewind $\mathcal{A}$ for the proofs of $\Pi_{\textsc{Bound}}$ to extract $\boldsymbol{s}_j$ for all dishonest parties. Change the output of $\mathcal{F}_{\textsc{Rand}}$ to achieve extraction.
6. Also rewind $\mathcal{A}$ to extract the witnesses from $\Pi_{\textsc{Lin}}$. If they do not match with the extracted $\boldsymbol{s}_j$ then abort.
7. Submit all the $\boldsymbol{s}_j$ of the dishonest parties to $\mathcal{F}_{\mathrm{KGD}}$ and obtain pk_i.
8. During Step (4) open each $C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_i)$ as pk_i by simulating $\Pi_{\textsc{Open-x}}$. Therefore fix the challenge in advance using $\mathcal{F}_{\textsc{Rand}}$.
9. For all dishonest parties in Step (4) also abort if the extracted witness for $C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_j)$ disagrees with the value pk_j announced by P_j.

Distributed Decryption:

1. The set of dishonest parties $\mathcal{I}$ is the same as in Π_{KG}. Let $\mathsf{sk}_i := \boldsymbol{F}_a^{\mathsf{sk}}\boldsymbol{s}_i$ and $C(\mathsf{sk}_i) = C(\boldsymbol{F}_a^{\mathsf{sk}}\boldsymbol{s}_i)$ be the same commitment as in the instance of Π_{KG}.
2. Sample $\boldsymbol{e}_i \xleftarrow{\$} \mathcal{U}_{\beta_d}^{l_r}$ for each honest P_i.
3. Compute the commitments $C(\boldsymbol{F}_c\mathsf{sk}_i), C(\boldsymbol{e}_i), C(\boldsymbol{F}_r\boldsymbol{e}_i)$ and $C(\boldsymbol{F}_c\mathsf{sk}_i + \boldsymbol{F}_r\boldsymbol{e}_i)$ honestly for all honest P_i, then broadcast them.
4. Run Step (3) honestly with the correct inputs for the honest parties.
5. In Step (3) use rewinding for the dishonest parties to extract the witnesses for $\Pi_{\textsc{Bound}}, \Pi_{\textsc{Lin}}, \Pi_{\textsc{Sum}}$. If a witness is not compatible with sk_j then abort. Also abort if the protocol aborts.
6. Rename the witnesses of the dishonest P_j from $\Pi_{\textsc{Sum}}$ as $\boldsymbol{d}_j$. Send these to $\mathcal{F}_{\mathrm{KGD}}$. Obtain $\boldsymbol{d}_i$ for all honest P_i from $\mathcal{F}_{\mathrm{KGD}}$.
7. In Step (4) simulate the opening of $C(\boldsymbol{F}_c\mathsf{sk}_i + \boldsymbol{F}_r\boldsymbol{e}_i)$ using $\Pi_{\textsc{Open-x}}$ as $\boldsymbol{d}_i$ by adjusting the output of $\mathcal{F}_{\textsc{Rand}}$.
8. For all dishonest parties in Step (4) abort if they prove that the value inside $C(\boldsymbol{F}_c\mathsf{sk}_j + \boldsymbol{F}_r\boldsymbol{e}_j)$ is different from $\boldsymbol{d}_j$ as extracted before.

Fig. 6. $\mathcal{S}_{\mathrm{KGD}}$: Simulator for the protocols $\Pi_{\mathrm{KG}}, \Pi_{\textsc{Dec}}$.

correct decryption key and succeeds using another one only by breaking the binding property of our scheme. We then define $\tau_{2,\textsc{Dec}}$ to be the same as $\tau_{1,\textsc{Dec}}$, just that the proofs in the simulation are now simulated by programming $\mathcal{F}_{\textsc{Rand}}$ appropriately, which yields $\tau_{1,\textsc{Dec}} \overset{s}{\approx} \tau_{2,\textsc{Dec}}$. Similarly as above, we define $\tau'_{1,\textsc{Dec}}$ to be the same as τ_{Π} where we now simulate the zero-knowledge proofs. This implies $\tau'_{1,\textsc{Dec}} \overset{s}{\approx} \tau_{\Pi}$. But observe that we can then again replace the commitments $C(\boldsymbol{e}_i)$, $C(\boldsymbol{F}_r\boldsymbol{e}_i)$, $C(\boldsymbol{F}_c\mathsf{sk}_i + \boldsymbol{F}_r\boldsymbol{e}_i)$ generated in Step (2) with those that were used in Step (3) of the Simulator $\mathcal{S}_{\mathrm{KGD}}$. Due to the statistical hiding property of $C(\cdot)$,

it follows that $\tau'_{1,\text{DEC}} \overset{s}{\approx} \tau'_{2,\text{DEC}}$. We now observe that $\tau'_{2,\text{DEC}} \overset{p}{\approx} \tau_{2,\text{DEC}}$ due to their construction, which concludes the proof. □

Optimizing away some of the Proofs. In practice we can, with a careful choice of parameters, avoid using the proof Π_{SUM}: opening the sum $C(a+b; \boldsymbol{r}_1 + \boldsymbol{r}_2) = C(a; \boldsymbol{r}_1) + C(b; \boldsymbol{r}_2)$ of two commitments $C(a; \boldsymbol{r}_1), C(b; \boldsymbol{r}_2)$ leaks information about the individual randomness $\boldsymbol{r}_1, \boldsymbol{r}_2$, of each commitment. This is why we use Π_{SUM} in the protocols to prove that a commitment opens to the sum of two other commitments.

On the other hand, if we open $C(a + b; \boldsymbol{r}_1 + \boldsymbol{r}_2)$ using $\Pi_{\text{OPEN-X}}$ then only $a+b$ is revealed, which does neither leak any information about the randomness of the sum nor of those of the individual terms. As an optimization one can therefore avoid the use of Π_{SUM} and simply add commitments directly, as long as the number of terms is small enough such that the randomness does not grow too large (which would break the binding of $C(\cdot)$).

D.3 Threshold Protocols for other Lattice-based Primitives

One might hope that the above techniques can also be used to give more efficient protocols for e.g. threshold signatures. There are (currently) two main approaches for lattice signatures, namely Fiat-Shamir style protocols like [Lyu09] or those that use a hash-and-sign approach such as [GPV08]. In the first case, such signature schemes have a rejection-sampling step where a bit is chosen with a certain abort probability that depends both on the signature and the secret basis, which is the signing key. This requires computation with very high precision. For hash-and-sign type constructions, the signer has to sample a short lattice vector using a trapdoor. It has been shown [BKP13] that this can actually be done in a distributed way, but their approach requires that all parties sample shares according to a Gaussian distribution. It is an interesting open question how to perform this efficiently with active security. For both cases, we do not see how our commitment scheme could be applied to solve the actual bottlenecks of the threshold versions, and leave this as an open problem.

E Extending the Threshold Protocols

For all practical purposes, the protocols $\Pi_{\text{KG}}, \Pi_{\text{DEC}}$ from Section 6 and Appendix D are not satisfactory. We will now improve them in multiple ways: in a first step, an extension to achieve UC security will be discussed. Moreover, we show a simple approach that allows to compute encryptions of powers of sk. This in turn can be used in an alternative distributed decryption algorithm that uses optimistic decryption. The protocols in this appendix are presented without proofs: their actual security depends on details of the schemes and chosen parameters which would complicate the presentation without yielding any new insights, and the basic structure of the protocols follows those from Section 6.

E.1 Some Further Assumptions

The starting point is to make some further assumptions about $D.\mathsf{Enc}$ from Definition 5. In Section 6, we only assumed that such an algorithm exists, while we now require that the encryption algorithm itself can be modeled in a similar way as $\mathsf{KG}, \mathsf{Dec}$ – namely, that it can be described in terms of linear operations.

Similarly to the message space R_q of $C(\cdot)$ we define the message space of Enc as R_p for $p \ll q$ (instead of $\{0,1\}^d$). By representing the coefficients of the elements as integers from the interval $(-p/2, p/2]$ we can naturally embed each $m \in R_p$ into $\mathbb{F}_q^N$. In particular, for a small enough number of ring operations in R_p we can simulate these operations on the embedding into $\mathbb{F}_q^N$, namely, for as long as the coefficients do not get too big and *wrap around modulo q*.

We assume that $l_e, \beta_e \in \mathbb{N}, \beta_e \ll q$ and N divides l_e. Similarly as before, l_e is the length of the randomness vector and β_e is the maximal norm of the randomness used in Enc, that we will now also describe in terms of linear operations: given a public key pk, we require that there exists a deterministic algorithm to compute two matrices $\boldsymbol{F}_e^{\mathsf{pk}} \in \mathbb{F}_q^{l_c \times l_e}, \boldsymbol{F}_e^m \in \mathbb{F}_q^{l_c \times N}$ which are independent of the plaintext and the noise, such that Enc, on an input $m \in R_p, \boldsymbol{e} \in \mathbb{F}_q^{l_e}$ with $||\boldsymbol{e}||_\infty \leq \beta_e$, performs the following operations:

$\mathsf{Enc}(\boldsymbol{F}_e^{\mathsf{pk}}, \boldsymbol{F}_e^m, m, \boldsymbol{e})$**:**

1. Consider the representation of m in $\mathbb{F}_q^N$, which we denote $\boldsymbol{m}$.
2. Compute $c = \boldsymbol{F}_e^{\mathsf{pk}}\boldsymbol{e} + \boldsymbol{F}_e^m\boldsymbol{m}$.
3. Output c.

In a nutshell, the above algorithm allows us to encrypt values we committed to *inside the commitment* without revealing the secret. A direct consequence of the above representation is that, if we assume the embedding of m to be homomorphic, that $\mathsf{Dec}_{\mathsf{sk}}(\mathsf{Enc}_{\mathsf{pk}}(m_1) + \mathsf{Enc}_{\mathsf{pk}}(m_2)) = m_1 + m_2$. Depending on the relationship between p, q and β_e, we may allow a number of such additions before Dec yields an incorrect value.

An additional requirement is that, given a ciphertext $c \in \mathbb{F}_q^{l_c}$ there exists a deterministic algorithm to compute $\boldsymbol{F}_c^m \in \mathbb{F}_q^{l_c \times N}, \boldsymbol{f}_c^m \in \mathbb{F}_q^{l_c}$ from c and independently of $a \in R_p$ such that $\mathsf{Dec}_{\mathsf{sk}}(c) \cdot a = \mathsf{Dec}_{\mathsf{sk}}(\boldsymbol{F}_c^m \cdot \boldsymbol{a} + \boldsymbol{f}_c^m)$ given the randomness in c is small enough. This a is a plaintext value, so we require that the result be decryptable with a normal secret key. Observe that publicly revealing a value $\boldsymbol{F}_c^m \cdot \boldsymbol{a} + \boldsymbol{f}_c^m$ may leak information on $\boldsymbol{a}$. We therefore *drown* the noise by adding new encryptions $\mathsf{Enc}_{\mathsf{pk}}(0)$ with noise bound β_m in the protocol. The details on the choice of all these parameters depend on the implementation of the protocols and are not discussed any further here. Similarly as above, we will require some additive homomorphism for a small number of additions of ciphertexts obtained from multiplication with a constant. This property follows from the linearity of the procedure for a suitable choice of parameters. An overview over the parameters and notation in this appendix can be found in Fig. 7.

Parameter	Explanation
l_e	Length of randomness vector for encryption
β_e	Noise bound for randomness in encryption
p	Modulus of plaintext space
R_p	Plaintext space of D
$\boldsymbol{F}_e^{\mathsf{pk}}$	Matrix applied to randomness vector in encryption
$\boldsymbol{F}_e^m$	Matrix applied to message in encryption
$\boldsymbol{F}_c^m, \boldsymbol{f}_c^m$	Values used to multiply ciphertext c by a constant
β_m	Noise bound for rerandomization of products with a constant
$\overline{\mathsf{pk}}_i, \overline{\mathsf{sk}}_i$	Public/private key pair of the party P_i

Fig. 7. Additional Parameters used in this Appendix.

E.2 Making the Protocols UC-secure

The proof of security crucially relies upon the simulator being able to extract witnesses from the ZK proofs by rewinding. Unfortunately, such rewinding is not possible within the UC framework. The standard workaround is to base the security on the simulator having other means for obtaining these values (e.g. having secret keys for some encryption scheme or a trapdoor for commitments). One then claims that a distinguisher between those two worlds exists and this distinguisher itself can then do rewinding (but will apparently not have access to the secret information of the simulator). In our case it is obvious that such a proof technique must fail, since we are not aware of trapdoors for our defined commitment scheme.

To make Π_{KG} UC-secure, we use the strengthened definition for the cryptosystem $D = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and make the additional (mild) setup assumption[11] that each party P_i has a key pair $(\overline{\mathsf{pk}}_i, \overline{\mathsf{sk}}_i)$ with publicly known $\overline{\mathsf{pk}}_i$.

The key generation protocol $\Pi_{\mathrm{KG,UC}}$ follows the same outline as Π_{KG}, with the one crucial difference: the seed $\boldsymbol{s}_i$ is sampled by party P_i in a special procedure `ProEncCommit` where it also generates an encryption $[\![\boldsymbol{s}_i]\!]$ under its key $\overline{\mathsf{pk}}_i$. P_i will publish the ciphertext and prove that it was computed correctly from $\boldsymbol{s}_i$ and some chosen randomness $\boldsymbol{e}$ using our zero-knowledge proofs. The simulator holds the keys $\overline{\mathsf{sk}}_i$ of the dishonest parties, is able to decrypt each ciphertext and can then send this value to $\mathcal{F}_{\mathrm{KGD}}$ as before. The protocol can be found in Fig. 8. A similar transformation can also be applied to Π_{DEC} and the remaining protocols from this appendix.

[11] Implicitly, in our protocol we further assume that $l_{\mathsf{pk}} = l_{\mathsf{sk}} = N$ to be able to encrypt public and private keys. This can easily be generalized, and we just make this assumption to simplify the exposition.

Protocol $\Pi_{\mathbf{KG,UC}}$

Procedure `ProEncCommit`(i)**:**

1. P_i locally samples $\boldsymbol{s} \xleftarrow{\$} \mathcal{U}^{l_s}_{\beta_s}$ as well as $\boldsymbol{e} \xleftarrow{\$} \mathcal{U}^{l_e}_{\beta_e}$ and computes $\boldsymbol{F}_e^{\overline{\mathsf{pk}}_i}\boldsymbol{e}, \boldsymbol{F}_e^m\boldsymbol{s}$.
2. P_i computes and broadcasts the commitments

$$C(\boldsymbol{s}), C(\boldsymbol{e}), C(\boldsymbol{F}_e^{\overline{\mathsf{pk}}_i}\boldsymbol{e}), C(\boldsymbol{F}_e^m\boldsymbol{s}) \text{ and } C(\boldsymbol{F}_e^{\overline{\mathsf{pk}}_i}\boldsymbol{e} + \boldsymbol{F}_e^m\boldsymbol{s})$$

   as well as $[\![\boldsymbol{s}]\!] = \boldsymbol{F}_e^{\overline{\mathsf{pk}}_i}\boldsymbol{e} + \boldsymbol{F}_e^m\boldsymbol{s}$.
3. Each P_i uses the following proofs towards all parties. Sample the challenge using $\mathcal{F}_{\text{RAND}}$:
   (a) Π_{BOUND} on $C(\boldsymbol{s})$ to show that $||\boldsymbol{s}||_\infty \leq \beta_s$.
   (b) Π_{BOUND} on $C(\boldsymbol{e})$ to show that $||\boldsymbol{e}||_\infty \leq \beta_e$.
   (c) Π_{LIN} on $C(\boldsymbol{e}), C(\boldsymbol{F}_e^{\overline{\mathsf{pk}}_i}\boldsymbol{e})$ using $\boldsymbol{F}_e^{\overline{\mathsf{pk}}_i}$.
   (d) Π_{LIN} on $C(\boldsymbol{s}), C(\boldsymbol{F}_e^m\boldsymbol{s})$ using $\boldsymbol{F}_e^m$.
   (e) Π_{SUM} on $C(\boldsymbol{F}_e^{\overline{\mathsf{pk}}_i}\boldsymbol{e}), C(\boldsymbol{F}_e^m\boldsymbol{s}), C(\boldsymbol{F}_e^{\overline{\mathsf{pk}}_i}\boldsymbol{e} + \boldsymbol{F}_e^m\boldsymbol{s})$.
   (f) $\Pi_{\text{OPEN-X}}$ on $C(\boldsymbol{F}_e^{\overline{\mathsf{pk}}_i}\boldsymbol{e} + \boldsymbol{F}_e^m\boldsymbol{s})$ to show that it opens to $[\![\boldsymbol{s}]\!]$.
   If any of the proofs fails, then abort.
4. Return $C(\boldsymbol{s})$.

Key Generation:

1. Each P_i runs $(\boldsymbol{s}_i, C(\boldsymbol{s}_i)) \leftarrow$ `ProEncCommit`(i).
2. Each P_i computes and broadcasts the commitments $C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_i), C(\boldsymbol{F}_a^{\mathsf{sk}}\boldsymbol{s}_i)$.
3. Each P_i uses the following proofs towards all parties. Sample the challenge using $\mathcal{F}_{\text{RAND}}$:
   (a) Π_{LIN} on $C(\boldsymbol{s}_i)$, $C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_i)$ using $\boldsymbol{F}_a^{\mathsf{pk}}$.
   (b) Π_{LIN} on $C(\boldsymbol{s}_i)$, $C(\boldsymbol{F}_a^{\mathsf{sk}}\boldsymbol{s}_i)$ using $\boldsymbol{F}_a^{\mathsf{sk}}$.
   If one of the proofs fails then abort.
4. Denote with pk_i the committed value in $C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_i)$. Each P_i proves to all parties that $C(\boldsymbol{F}_a^{\mathsf{pk}}\boldsymbol{s}_i)$ contains pk_i using $\Pi_{\text{OPEN-X}}$. If one of the proofs fails, then abort.
5. If all proofs were correct, then output $\mathsf{pk} = \sum_{i\in[n]} \mathsf{pk}_i$.

Fig. 8. $\Pi_{\text{KG,UC}}$: Protocol for actively secure key generation with UC security.

E.3 Computing Powers of the Secret Key

We can moreover use the additional assumptions made on D to allow the computation of powers of the key sk securely. It may first seem counter-intuitive why one would want to compute such a value, but the reason lies in potential homomorphic properties of D:

- The encryption scheme due to [BV11] has an inherent ciphertext growth due to multiplications. The actual key that is used in decryption consists of powers of the secret key. To allow distributed decryption, a sharing of such a power of a secret key must be computed.

– The [BGV12] cryptosystem uses a *key switching matrix* to cope with the ciphertext growth of [BV11], but this matrix is computed as an encryption of sk^2 (times some constant).

This task of computing a power of the secret key can be achieved using our commitment scheme, its protocols and D. In a proof of security for our protocol, one would have to make an additional assumption on D, namely that it is KDM-secure [BRS02].

Here is how the protocol $\Pi_{\text{KeySquared}}$ works on an intuitive level: first, observe that there already are commitments to each sk_i from Π_{KG}. These commitments can be used in a first step to compute an encryption $c = \mathsf{Enc}_{\mathsf{pk}}(\mathsf{sk})$ of the secret key under its public key. This is possible because we can encrypt values that were contained in a commitment into correct ciphertexts, something which we already did in $\Pi_{\text{KG,UC}}$. Therefore, P_i will encrypt sk_i and prove correctness of the ciphertext. It is safe to reveal this encryption due to the KDM assumption on the cryptosystem. After this is done, these ciphertexts can be added up to obtain c.

Now observe that each share sk_i can be considered as a plaintext element, so we can multiply them with c. This can be done if we compute the matrices $\boldsymbol{F}_c^m, \boldsymbol{f}_c^m$ which must exist by assumption on the cryptosystem. These matrices are public and applied to each $C(\mathsf{sk}_i)$ individually, where each P_i knows the correct value that opens the resulting commitment. Before opening it, each P_i will rerandomize the resulting ciphertext such as to hide the share sk_i. The result of the protocol as depicted in Fig. 9 is then an encryption of sk^2 under pk.

E.4 An Alternative Solution to Distributed Decryption

We want to point out that an alternative approach for distributed decryption can be based on *optimistic decryption*, where the zero-knowledge proofs for the commitments are only executed in the case of a discovered decryption failure (to uncover a dishonest party). During a regular protocol run we will rely on proofs of plaintext knowledge for the ciphertexts which can be amortized using the technique from [BDLN16,CDXY17]. The reliable decryption technique is similar to [LSS16], but we moreover allow to identify the cheater.

The optimistic decryption requires that D is somewhat homomorphic. We require that there exists an algorithm $\otimes$ that allows to multiply ciphertexts in a way that allows decryption using $D.\mathsf{Dec}$. Such an algorithm can be realized using the output of $\Pi_{\text{KeySquared}}$.

Definition 6 (Multiplicative Property). *A distributed cryptosystem D is said to have the* multiplicative property *if there exists a deterministic* $\mathsf{poly}(\lambda)$*-time algorithm $\otimes$ such that*

$$\Pr\left[m \neq a \cdot b \;\middle|\; \begin{array}{l} (\mathsf{pk},\mathsf{sk}) \leftarrow \mathsf{KG}(1^\lambda) \wedge a, b \in R_p \wedge \\ c_a \leftarrow \mathsf{Enc}_{\mathsf{pk}}(a) \wedge c_b \leftarrow \mathsf{Enc}_{\mathsf{pk}}(b) \wedge \\ c \leftarrow c_a \otimes c_b \wedge m \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c) \end{array} \right] \leq \mathsf{negl}(\lambda),$$

Protocol $\Pi_{\mathbf{KeySquared}}$

We assume that a commitment $C(\mathsf{sk}_i)$ of each secret key share is available from Π_{KG} and that $||\mathsf{sk}_i||_\infty < p$.

1. Each P_i samples $\boldsymbol{v}_i \xleftarrow{\$} \mathcal{U}^{l_e}_{\beta_e}$ and computes and broadcasts the commitments

   $$C(\boldsymbol{v}_i), C(\boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{v}_i), C(\boldsymbol{F}^m_e \mathsf{sk}_i), C(\boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{v}_i + \boldsymbol{F}^m_e \mathsf{sk}_i)$$ as well as $[\![\mathsf{sk}_i]\!] = \boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{v}_i + \boldsymbol{F}^m_e \mathsf{sk}_i$.

2. Each P_i uses the following proofs towards all parties. Sample the challenge using $\mathcal{F}_{\text{RAND}}$:
   (a) Π_{BOUND} on $C(\boldsymbol{v}_i)$ to show that $||\boldsymbol{v}_i||_\infty \leq \beta_e$.
   (b) Π_{LIN} on $C(\boldsymbol{e}_i)$, $C(\boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{v}_i)$ using $\boldsymbol{F}^{\mathsf{pk}}_e$.
   (c) Π_{LIN} on $C(\mathsf{sk}_i)$, $C(\boldsymbol{F}^m_e \mathsf{sk}_i)$ using $\boldsymbol{F}^m_e$.
   (d) Π_{SUM} on $C(\boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{v}_i), C(\boldsymbol{F}^m_e \mathsf{sk}_i), C(\boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{v}_i + \boldsymbol{F}^m_e \mathsf{sk}_i)$.
   (e) $\Pi_{\text{OPEN-X}}$ on $C(\boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{v}_i + \boldsymbol{F}^m_e \mathsf{sk}_i)$ to show that it opens to $[\![\mathsf{sk}_i]\!]$.
   If one of the proofs fails then abort.
3. Each P_i locally computes $[\![\mathsf{sk}]\!] = \sum_{j=1}^n [\![\mathsf{sk}_j]\!]$ and $\boldsymbol{F}^m_c, \boldsymbol{f}^m_c$ from $[\![\mathsf{sk}]\!]$.
4. Each P_i samples $\boldsymbol{w}_i \xleftarrow{\$} \mathcal{U}^{l_e}_{\beta_m}$ and computes and broadcasts the commitments

   $$C(\boldsymbol{w}_i), C(\boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{w}_i), C(\boldsymbol{F}^m_c \mathsf{sk}_i + \delta_{1i} \cdot \boldsymbol{f}^m_c), C(\boldsymbol{F}^m_c \mathsf{sk}_i + \delta_{1i} \cdot \boldsymbol{f}^m_c + \boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{w}_i),$$

   as well as

   $$[\![\mathsf{sk} \cdot \mathsf{sk}_i]\!] = \boldsymbol{F}^m_c \mathsf{sk}_i + \delta_{1i} \cdot \boldsymbol{f}^m_c + \boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{w}_i$$

   where δ_{xy} is the Kronecker Delta function.
5. Each P_i uses the following proofs towards all parties. Sample the challenge using $\mathcal{F}_{\text{RAND}}$:
   (a) Π_{BOUND} on $C(\boldsymbol{w}_i)$ to show that $||\boldsymbol{w}_i||_\infty \leq \beta_m$.
   (b) Π_{LIN} on $C(\boldsymbol{w}_i), C(\boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{w}_i)$ using $\boldsymbol{F}^{\mathsf{pk}}_e$.
   (c) Π_{LIN} on $C(\mathsf{sk}_i), C(\boldsymbol{F}^m_c \mathsf{sk}_i + \delta_{1i} \cdot \boldsymbol{f}^m_c)$ using the linear function $g(x) = \boldsymbol{F}^m_c x + \delta_{1i} \cdot \boldsymbol{f}^m_c$.
   (d) Π_{SUM} on $C(\boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{w}_i), C(\boldsymbol{F}^m_c \mathsf{sk}_i + \delta_{1i} \cdot \boldsymbol{f}^m_c), C(\boldsymbol{F}^m_c \mathsf{sk}_i + \boldsymbol{f}^m_c + \boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{w}_i)$.
   (e) $\Pi_{\text{OPEN-X}}$ on $C(\boldsymbol{F}^m_c \mathsf{sk}_i + \delta_{1i} \cdot \boldsymbol{f}^m_c + \boldsymbol{F}^{\mathsf{pk}}_e \boldsymbol{w}_i)$ to show that it opens to $[\![\mathsf{sk} \cdot \mathsf{sk}_i]\!]$.
   If one of the proofs fails then abort.
6. Each P_i locally computes $[\![\mathsf{sk}^2]\!] = \sum_{j=1}^n [\![\mathsf{sk} \cdot \mathsf{sk}_j]\!]$. Output $[\![\mathsf{sk}^2]\!]$.

Fig. 9. $\Pi_{\text{KEYSQUARED}}$: Protocol for actively secure generation of powers of secret keys.

where the randomness is taken over[12] *the choice of inputs for* $\mathsf{KG}, \mathsf{Dec}, \mathsf{Enc}$.

Similarly as for $\Pi_{\text{KEYSQUARED}}$ we will not prove the security of the protocol, but give some intuition on how it works: to decrypt a ciphertext $[\![x]\!]$ the parties first generate an encryption of a uniformly random value a. They then *encode* x by computing the product $[\![b]\!] = [\![x]\!] \otimes [\![a]\!]$. Before decrypting all three ciphertexts unreliably, each P_i commits to the values $\boldsymbol{F}_x \mathsf{sk}_i, \boldsymbol{e}^x_i, \boldsymbol{F}_r \boldsymbol{e}^x_i, \boldsymbol{d}^x_i$ that it will use in Dec to decrypt $[\![x]\!]$, as well as those values used in the decryption

[12] To ease of readability, we leave out the full specification of the inputs to $\mathsf{KG}, \mathsf{Dec}$ but simply assume that they are correct according to $\mathcal{F}_{\text{KGD}}$.

Protocol $\Pi_{\mathbf{DecAlt}}$

A protocol to decrypt a ciphertext $[\![x]\!]$.

1. Each P_i samples $a_i \xleftarrow{\$} R_p$ uniformly at random and computes $[\![a_i]\!] \leftarrow \mathsf{Enc}_{\mathsf{pk}}(a_i)$.
2. Each P_i broadcasts $[\![a_i]\!]$ together with a proof of plaintext knowledge for Enc.
3. Each P_i locally computes $[\![a]\!] = \sum_{i\in[n]}[\![a_i]\!]$ and $[\![b]\!] \leftarrow [\![x]\!] \otimes [\![a]\!]$.
4. Each P_i locally samples randomness $\boldsymbol{e}_i^x, \boldsymbol{e}_i^a, \boldsymbol{e}_i^b \xleftarrow{\$} \mathcal{U}_{\beta_d}^{l_r}$ and computes $\boldsymbol{F}_x$ as used in Dec to decrypt $[\![x]\!]$ as well as $\boldsymbol{F}_a$ for $[\![a]\!]$ and $\boldsymbol{F}_b$ for $[\![b]\!]$. Then set
$$\boldsymbol{d}_i^x = \boldsymbol{F}_x\mathsf{sk}_i + \boldsymbol{F}_r\boldsymbol{e}_i^x \text{ and } \boldsymbol{d}_i^a = \boldsymbol{F}_a\mathsf{sk}_i + \boldsymbol{F}_r\boldsymbol{e}_i^a \text{ and } \boldsymbol{d}_i^b = \boldsymbol{F}_b\mathsf{sk}_i + \boldsymbol{F}_r\boldsymbol{e}_i^b.$$
5. Each P_i broadcasts
$$C(\boldsymbol{F}_x\mathsf{sk}_i), C(\boldsymbol{F}_a\mathsf{sk}_i), C(\boldsymbol{F}_b\mathsf{sk}_i), C(\boldsymbol{e}_i^x), C(\boldsymbol{e}_i^a), C(\boldsymbol{e}_i^b) \text{ as well as}$$
$$C(\boldsymbol{F}_r\boldsymbol{e}_i^x), C(\boldsymbol{F}_r\boldsymbol{e}_i^a), C(\boldsymbol{F}_r\boldsymbol{e}_i^b), C(\boldsymbol{d}_i^x), C(\boldsymbol{d}_i^a), C(\boldsymbol{d}_i^b).$$
6. Each P_i generates auxiliary commitments to commit to $\boldsymbol{d}_i^x, \boldsymbol{d}_i^a, \boldsymbol{d}_i^b$ towards all parties.
7. Each P_i opens the auxiliary commitments to $\boldsymbol{d}_i^x, \boldsymbol{d}_i^a, \boldsymbol{d}_i^b$.
8. All parties check that
$$\mathsf{decode}(\sum_{i\in[n]} \boldsymbol{d}_i^x) \cdot \mathsf{decode}(\sum_{i\in[n]} \boldsymbol{d}_i^a) = \mathsf{decode}(\sum_{i\in[n]} \boldsymbol{d}_i^b).$$
If yes, then they output $x \leftarrow \mathsf{decode}(\sum_{i\in[n]} \boldsymbol{d}_i^x)$ and terminate.
9. Otherwise, for each $i \in [n]$ the parties do the following, where all parties abort with P_i if a check fails:
   (a) P_i proves using Π_{BOUND} that $C(\boldsymbol{e}_i^x), C(\boldsymbol{e}_i^a), C(\boldsymbol{e}_i^b)$ have ∞-norm at most β_d.
   (b) P_i proves using Π_{LIN} that $C(\boldsymbol{F}_x\mathsf{sk}_i), C(\boldsymbol{F}_a\mathsf{sk}_i), C(\boldsymbol{F}_b\mathsf{sk}_i)$ are derived from $C(\mathsf{sk}_i)$ using $\boldsymbol{F}_x, \boldsymbol{F}_a, \boldsymbol{F}_b$ and that $C(\boldsymbol{F}_r\boldsymbol{e}_i^x), C(\boldsymbol{F}_r\boldsymbol{e}_i^a), C(\boldsymbol{F}_r\boldsymbol{e}_i^b)$ are derived from $C(\boldsymbol{e}_i^x), C(\boldsymbol{e}_i^a), C(\boldsymbol{e}_i^b)$ using $\boldsymbol{F}_r$.
   (c) P_i runs Π_{SUM} on the tuples
   - $(C(\boldsymbol{F}_x\mathsf{sk}_i), C(\boldsymbol{F}_r\boldsymbol{e}_i^x), C(\boldsymbol{d}_i^x))$
   - $(C(\boldsymbol{F}_a\mathsf{sk}_i), C(\boldsymbol{F}_r\boldsymbol{e}_i^a), C(\boldsymbol{d}_i^a))$
   - $(C(\boldsymbol{F}_b\mathsf{sk}_i), C(\boldsymbol{F}_r\boldsymbol{e}_i^b), C(\boldsymbol{d}_i^b))$.

   (d) P_i proves using $\Pi_{\text{OPEN-X}}$ that $C(\boldsymbol{d}_i^x)$, $C(\boldsymbol{d}_i^a)$, $C(\boldsymbol{d}_i^b)$ open to $\boldsymbol{d}_i^x, \boldsymbol{d}_i^a, \boldsymbol{d}_i^b$.

Fig. 10. Π_{DECALT}: Alternative protocol for the decryption of ciphertexts.

of $[\![a]\!], [\![b]\!]$. Thereafter, the parties unreliably decrypt $[\![x]\!], [\![a]\!], [\![b]\!]$ by opening the commitments to $\boldsymbol{d}_i^x, \boldsymbol{d}_i^a$ and $\boldsymbol{d}_i^b$. All parties check that $a \cdot x = b$.

If this equality holds, then we consider the result as correct. If, on the other hand, it does not hold, then each party proves in zero-knowledge that its $\boldsymbol{d}_i^x, \boldsymbol{d}_i^a, \boldsymbol{d}_i^b$ were correctly generated (as in a correct decryption procedure) based on the commitments that it provided. The protocol is presented in Fig. 10, where Step (1) and Step (2) can be done ahead of decryption time.