An Unconditionally Hiding and Long-Term Binding Post-Quantum Commitment Scheme

Daniel Cabarcas[1], Denise Demirel[2], Florian Göpfert[3], Jean Lancrenon[4], and Thomas Wunderer[5]

[1] National University of Colombia, Columbia, cabarcas@gmail.com
[2] Technische Universität Darmstadt, Germany, ddemirel@cdc.informatik.tu-darmstadt.de
[3] Technische Universität Darmstadt, Germany, fgoepfert@cdc.informatik.tu-darmstadt.de
[4] University of Luxembourg, Luxembourg, jean.lancrenon@uni.lu
[5] Technische Universität Darmstadt, Germany, twunderer@cdc.informatik.tu-darmstadt.de

Abstract. Commitment schemes are among cryptography's most important building blocks. Besides their basic properties, hidingness and bindingness, for many applications it is important that the schemes applied support proofs of knowledge. However, all existing solutions which have been proven to provide these protocols are only computationally hiding or are not resistant against quantum adversaries. This is not suitable for long-lived systems, such as long-term archives, where commitments have to provide security also in the long run. Thus, in this work we present a new post-quantum unconditionally hiding commitment scheme that supports (statistical) zero-knowledge protocols and allows to refreshes the binding property over time. The bindingness of our construction relies on the approximate shortest vector problem, a lattice problem which is conjectured to be hard for polynomial approximation factors, even for a quantum adversary. Furthermore, we provide a protocol that allows the committer to prolong the bindingness property of a given commitment while showing in zero-knowledge fashion that the value committed to did not change. In addition, our construction yields two more interesting features: one is the ability to "convert" a Pedersen commitment into a lattice-based one, and the other one is the construction of a hybrid approach whose bindingness relies on the discrete logarithm and approximate shortest vector problems.

Keywords: unconditionally hiding commitments, post-quantum, lattice-based cryptography, long-term security, proof of knowledge

1 Introduction

Commitment schemes are arguably one of cryptography's most basic and important primitives. Indeed, they can be found as building blocks within many

cryptographic concepts and services, such as zero-knowledge protocols [6, 8], secret sharing [22], key exchange [5], and others. A commitment scheme allows one to publicly commit to a hidden value which may be later revealed. It is required that 1) once the value has been committed to, it can no longer be changed by its owner, and 2) the value remains hidden until the owner chooses to reveal it. A scheme possessing the first property is said to be *binding*, and one with the second property is called *hiding*.

Especially for long-lived systems such as long-term archiving (e.g., [14]), one would like these security properties to both hold *perfectly*, i.e. even in the presence of computationally unbounded adversaries. However, it is well-known (and easy to see) that this is impossible. Thus, the best one can hope for is a scheme that is *computationally binding* and *unconditionally (i.e. statistically or perfectly) hiding*, or vice-versa. In other words, given a fixed commitment scheme, at least one of these two properties will fail for this scheme, given enough time.

In addition to the basic properties (bindingness and hidingness), many applications require commitment schemes with additional functionalities. Arguably one of the most important among those are proof of knowledge protocols. They allow, for instance, a committer to convince a challenger that it can open a commitment, without revealing any additional information about the opening value or secret.

Benhamouda et al. [4] presented an interesting lattice-based commitment scheme supporting proofs of knowledge. The authors show that the scheme is unconditionally binding and computationally hiding even for a polynomial time *quantum* adversary. This scheme can consequently be considered as post-quantum secure. Nevertheless, since it is only computationally hiding, it can not be used in a long-term setting processing secret data. This is due to the fact that an attacker can simply store the commitment, wait until it has access to enough resources to solve the computational problem providing the hiding property, and recover the secret. Therefore, long-term secure commitment schemes must provide unconditional hidingness from the very beginning. In addition, since in a long-term setting it is not reasonable to dismiss the future existence of efficient quantum computers, the schemes must also be post-quantum secure. Developing an unconditionally hiding commitment scheme supporting proofs of knowledge provides another valuable property for long-lived systems. They allow to prolong the computational binding property over time while allowing the committer to prove in zero-knowledge fashion that the value committed to did not change. How to achieve this for the Pedersen commitment scheme has been shown by Demirel et al. [10]. In this work we show how this can be transferred to the post-quantum world.

1.1 Contribution and Roadmap

In this work we present LPCom, a **L**ong-term **P**ost-quantum **Com**mitment scheme. Our construction can be seen as a modification of the scheme presented by Benhamouda et al. [4]. To the best of our knowledge LPCom is the first unconditionally hiding and computationally binding lattice-based commitment

scheme, which so far has been proven to support proofs of knowledge and offers prolongable bindingness.

In Section 3 we introduce LPCom and prove that correctly instantiated it is unconditionally hiding and computationally binding under standard lattice assumptions. Section 4 shows how to perform proof of knowledge protocols for LPCom. These protocols are used in Section 5 to construct proof of message equality protocols, from which we build a protocol for prolonging the bindingness of LPCom. Finally, in Section 6 we discuss how the proofs of message equality can also be used to convert a Pedersen commitment into an LPCom commitment, allowing to provide double-hardness (i.e. the security is based on two different hardness assumptions). This is desired in long-term systems, since it can prevent a loss in security in case of a sudden breakthrough on cryptanalysis.

2 Background and Notation

2.1 General Notation

Throughout the paper, vectors will be denoted by small bold letters (e.g. $\boldsymbol{v}$), while matrices will be denoted by capital bold letters (e.g. $\boldsymbol{A}$). We will write $\|\boldsymbol{v}\|$ for the euclidian norm of the vector $\boldsymbol{v}$. Furthermore, for $q \in \mathbb{Z}$ we write $\mathbb{Z}_q$ instead of $\mathbb{Z}/q\mathbb{Z}$. Sometimes, abusing notation, we identify elements in $\mathbb{Z}_q$ with elements in $\mathbb{Z}$. For this, we will identify a residue class in $\mathbb{Z}_q$ with its representative of smallest absolute value whenever necessary. We also extend this identification coordinate-wise to elements of $\mathbb{Z}_q^n$. It should always be clear from the context when an element in $\mathbb{Z}_q$ is viewed as an element in $\mathbb{Z}$. Using this notation, we define the length $\|\boldsymbol{v}\|$ for an element $\boldsymbol{v} \in \mathbb{Z}_q^n$, by taking the norm of the corresponding element in $\mathbb{Z}^n$. Notice that this length function still satisfies the triangular inequality.

A function $\mathrm{f} : \mathbb{Z} \to \mathbb{R}$ is said to be negligible (in x) if for every positive integer c, we have $|\mathrm{f}(x)| < 1/x^c$ for every sufficiently large x. Based on this notation, a (probability) function $p : \mathbb{Z} \to [0, 1]$ is called overwhelming (in x) if $1 - p$ is negligible (in x). The statistical distance is a measure for the difference of two probability distributions. For a precise definition we refer to [11].

2.2 Lattices

A subset $\Lambda \subset \mathbb{R}^n$ is called a lattice in $\mathbb{R}^n$ if there exists an $m \in [0, n]$ and m $\mathbb{R}$-linearly independent vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ such that $\Lambda = \mathbb{Z}\boldsymbol{b}_1 + \ldots + \mathbb{Z}\boldsymbol{b}_m$. In this case, the vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ are called a basis of the lattice Λ. The fundamental parallelepiped of a basis is defined as $\mathcal{P} = \{\sum_{i=1}^m \alpha_i \boldsymbol{b}_i \mid \alpha_i \in [0, 1)\}$. The dimension $\dim(\Lambda)$ of a lattice Λ is denoted by the maximal number of $\mathbb{R}$-linearly independent lattice vectors. A lattice $\Lambda \subseteq \mathbb{R}^n$ has full rank if $\dim(\Lambda) = n$. A lattice $\Lambda \subseteq \mathbb{R}^n$ is called an integer lattice if $\Lambda \subset \mathbb{Z}^n$. The determinant $\det(\Lambda)$ of a full rank lattice Λ is defined as the n-dimensional volume of the fundamental parallelepiped. Note that the determinant of the lattice does not depend on the

basis chosen but only on the lattice itself. A lattice $\Lambda \subseteq \mathbb{R}^n$ is called q-ary for some integer q if it is an integer lattice containing $q\mathbb{Z}^n$ as a sublattice. In the following, we will only be dealing with q-ary lattices, which by definition are full ranked. Typically, a q-ary lattice is given in form of a matrix $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$. More precisely, for a matrix $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$ we define the q-ary lattice

$$\Lambda_q(\boldsymbol{A}) = \{\boldsymbol{y} \in \mathbb{Z}^m \mid \exists x \in \mathbb{Z}_q^n : \boldsymbol{A}\boldsymbol{x} = \boldsymbol{y} \mod q\}.$$

While algebraic problems related to lattices (such as calculating intersections and sums of lattices or checking whether a certain vector is a lattice vector) are typically easy, certain geometric problems are believed to be hard, even in the presence of quantum computers. The arguably most important lattice problem is the *Shortest Vector Problem* (SVP): Given a lattice Λ, find a shortest non-zero lattice vector. The length of a shortest non-zero lattice vector is referred to as $\lambda_1(\Lambda)$. More general, for a lattice $\Lambda \subset \mathbb{R}^n$ and $i \in \mathbb{N}$, by $\lambda_i(\Lambda)$ we denote the radius of the smallest n-dimensional ball centered around the origin that contains at least i $\mathbb{R}$-linearly independent lattice vectors. We call $\lambda_i(\Lambda)$ the i-th successive minimum of Λ.
The Gaussian heuristic is typically considered a good tool for estimating the size of the successive minima of a lattice and frequently used in lattice-based cryptography. In its original form, it estimates the number of lattice points in a given set. We give a version of it that is used to estimate the last successive minimum of a lattice.

Assumption 1 (Gaussian Heuristic) *The m-th successive minimum of an m-dimensional lattice Λ can be approximated by*

$$\lambda_m(\Lambda) \approx \left(\frac{\det(\Lambda)}{V_m}\right)^{1/m},$$

where V_m is the volume of an m-dimensional ball with radius one.

For m big enough, we have

$$V_m^{1/m} = \left(\frac{\pi^{m/2}}{\Gamma(m/2+1)}\right)^{1/m} \geq \frac{\pi^{1/2}}{\left((m/2)^{m/2}\right)^{1/m}} = \sqrt{\frac{2\pi}{m}}.$$

Combining this with the Gaussian heuristic leads to the assumption

$$\lambda_m(\Lambda) \leq \sqrt{\frac{m}{2\pi}} \det(\Lambda)^{1/m}. \tag{1}$$

Rather than the SVP, the hardness of lattice-based schemes is typically based on a relaxation denoted by α-SVP: Given a lattice Λ and some $\alpha \geq 1$, find a non-zero lattice vector of norm at most $\alpha\lambda_1(\Lambda)$. This problem is conjectured to be hard as long as α is polynomial in the lattice dimension and the lattice contains no trivial non-zero vectors (in a q-ary lattice, we say vectors in $q\mathbb{Z}^n$ are trivial) of length at most $\alpha\lambda_1(\Lambda)$.

One might ask how hard finding non-trivial short lattice vectors is in practice. Gama and Nguyen [12] identified the hermite factor δ as the main parameter determining the hardness of finding a lattice vector shorter than a given length. The hermite factor δ of a lattice vector $\boldsymbol{v}$ in an m-dimensional lattice Λ is given by

$$\delta^m = \frac{\det(\Lambda)^{1/m}}{\|\boldsymbol{v}\|}.$$

Current estimates claim that finding vectors with hermite factor $\delta = 1.01$ is hard, and already slightly smaller hermite deltas are considered out of reach for a foreseeable future. Let Λ be a q-ary lattice and q be minimal with this property. Then the trivial vectors are $q\mathbb{Z}^n$, and therefore we can conclude that α-SVP on Λ is considered to be hard as long as α is polynomial in the lattice dimension and $\alpha\lambda_1(\Lambda) < q$.

Lattice-based cryptographic schemes often make use of values that are sampled according to a discrete Gaussian distribution. Recall that the continuous Gaussian distribution $\overline{D}_\sigma$ with Gaussian parameter σ is defined by its density function $\rho_\sigma(x) = \exp(-\pi x^2/\sigma^2)/\sigma$. Likewise, the n-dimensional discrete Gaussian distributions $D^n_{\boldsymbol{v},\sigma}$ centered around some $\boldsymbol{v} \in \mathbb{R}^n$ with Gaussian parameter σ is defined by

$$\Pr[D^n_{\boldsymbol{v},\sigma} = \boldsymbol{x}] = \frac{\rho_\sigma(\|\boldsymbol{x} - \boldsymbol{v}\|)}{\sum_{\boldsymbol{w}\in\mathbb{Z}^n} \rho_\sigma(\|\boldsymbol{w}\|)}.$$

For a probability distribution D on a set X we use the notation $x \xleftarrow{\$} D$ in order to say that x is sampled according to D. If x is sampled uniformly on a set X, we simply write $x \xleftarrow{\$} X$.

2.3 Commitment Schemes

A commitment scheme consists of three probabilistic polynomial-time (PPT) algorithms $(\mathsf{GenCom}, \mathsf{Com}, \mathsf{Unv})$ of the following form.

$\mathsf{GenCom}(1^\kappa, 1^k)$ The *generation algorithm* GenCom takes as input 1^κ and 1^k, for security parameters κ and k, and outputs a public commitment key pk. Note that pk defines a message space $\mathcal{M}$.

$\mathsf{Com}(pk, v)$ The *commitment algorithm* Com takes as input a value $v \in \mathcal{M}$ and a commitment key pk and outputs a commitment c and an opening value r.

$\mathsf{Unv}(pk, c, v, r)$ The *unveil algorithm* takes as input a commitment key pk, a value v, a commitment c, and an opening value r, and returns v or $\perp$.

We will suppose implicitly the presence of pk in the remainder, omitting it in the notation.

A commitment scheme is called:

Correct if for all $v \in \mathcal{M}$, the *unveil algorithm* returns v with overwhelming probability whenever the inputs were computed honestly, i.e.,

$$\Pr[\mathsf{Unv}(pk, c, v, r) = v : (c, r) \leftarrow \mathsf{Com}(pk, v)] = 1 - \varepsilon.$$

for some ε negligible in k.

Statistically Hiding if for any pair $v, v' \in \mathcal{M}$ the distribution of the randomized commitments output by $\mathsf{Com}(pk, v)$ and $\mathsf{Com}(pk, v')$ is statistically close in κ.

Computationally Binding if for any probabilistic polynomial-time adversary P, the probability to find pairs of opening values (v, r) and (v', r') with $v \neq v'$ to the same commitment value c that get accepted by the unveil algorithm is negligible, i.e.,

$$\Pr[\mathsf{Unv}(pk, c, v, r) = v \wedge \mathsf{Unv}(pk, c, v', r') = v' \wedge \boldsymbol{v} \neq \boldsymbol{v}' : (c, v, r, v', r') \leftarrow P(pk)] \leq \varepsilon$$

for some ε negligible in k.

In addition we want to have the following property.

Prolongable Bindingness: A commitment scheme provides prolongable bindingness if for any commitment $c_1 = \mathsf{Com}_1(pk_1, v)$ a polynomial time bounded party that knows the opening values can generate a new commitment $c_2 = \mathsf{Com}_2(pk_2, v)$ with a higher security level, such that it can prove in statistical zero-knowledge fashion that it can open both commitments to the same value v.

While statistical and perfect hidingness are typically considered equivalent, the situation is not so easy when commitments must be valid for a long time. While perfectly hiding schemes do not provide any information abut the message, statistically hiding schemes allow an attacker to gain a negligible amount of information. One might think that an attacker can store the old commitments, wait until it has access to enough computational resources and use those resources to amplify this information. However, this would require him to run a big amount of independent attacks, which is impossible since it has only access to a polynomial number of commitments.

Our scheme is crafted such that κ allows to bound the information that can be extracted by an outside attacker. Note that this is independent of the underlying computational hardness guaranteeing the binding property tuned by the security parameter k.

3 LPCom: A new Lattice-Based Commitment Scheme

In this section, we introduce our lattice-based commitment scheme LPCom (**L**ong-term **P**ost-quantum **Com**mitment scheme). Note that our construction is similar to the unconditionally binding commitment scheme proposed by Benhamouda et al. in [4]. However, besides the fact that their commitment does not provide hidingness unconditionally it relies on a different hardness assumption. After giving the definition of LPCom in Figure 1 of Section 3.1, Section 3.2 provides conditions on the chosen parameters such that LPCom is statistically hiding and computationally binding. In Figure 2 of Section 3.3 we propose parameters for LPCom and prove that, instantiated with this parameter set, LPCom is statistically hiding and computationally binding.

3.1 Description of the Scheme

Figure 1 describes our commitment scheme LPCom. The generation algorithm GenCom chooses a lattice dimension m, a message space rank n, a modulus q, a Gaussian parameter σ, and an error bound B. It then samples two matrices $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ defining the commitment function, and sets a message space $\mathcal{M}$.

The commitment algorithm Com samples a randomness value $\boldsymbol{r}$ and a Gaussian error term $\boldsymbol{e}$ and computes the commitment corresponding to the value and those randomizers.

The unveil algorithm Unv checks if the error term used in the commitment is small enough (i.e. smaller than B) and returns the value if and only if this is the case.

$\mathsf{GenCom}(1^\kappa, 1^k)$ Set appropriate parameters $n, m, q \in \mathbb{Z}$ and $B, \sigma \in \mathbb{R}^+$ and sample two matrices $\boldsymbol{A}_1 \xleftarrow{\$} \mathbb{Z}_q^{m\times n}$, $\boldsymbol{A}_2 \xleftarrow{\$} \mathbb{Z}_q^{m\times k}$ such that $\boldsymbol{A} = \begin{pmatrix}\boldsymbol{A}_1 & \boldsymbol{A}_2\end{pmatrix} \in \mathbb{Z}_q^{m\times(n+k)}$ has trivial kernel. Furthermore, define the message space $\mathcal{M}$ as a subgroup of $\mathbb{Z}_q^n$.

$\mathsf{Com}(\boldsymbol{v} \in \mathcal{M})$ Sample $\boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_q^k$ and $\boldsymbol{e} \xleftarrow{\$} D_\sigma^m$, calculate

$$\boldsymbol{c} = \mathsf{com}(\boldsymbol{v}, \boldsymbol{r}, \boldsymbol{e}) = \boldsymbol{A}_1\boldsymbol{v} + \boldsymbol{A}_2\boldsymbol{r} + \boldsymbol{e} \in \mathbb{Z}_q^m,$$

and output $\boldsymbol{c}, \boldsymbol{r}$.

$\mathsf{Unv}(\boldsymbol{c}, \boldsymbol{v}, \boldsymbol{r})$ Return $\boldsymbol{v}$ if $\|\boldsymbol{c} - \boldsymbol{A}_1\boldsymbol{v} - \boldsymbol{A}_2\boldsymbol{r}\| \leq B$, and $\perp$ if not.

Fig. 1. LPCom

3.2 Correctness, Hidingness, and Bindingness

The unveil function obviously accepts an honestly created commitment as long as the sampled error term $\boldsymbol{e}$ has norm less or equal than B. Since technically the Gaussian distribution D_σ^m is not bounded, in theory it might happen that an honestly created commitment gets rejected by the unveil funtion. However, the following theorem provides a condition on the parameters such that the chance of this happening is negligible.

Theorem 1 (correctness) *Let $n, k, m, q \in \mathbb{Z}$, $\sigma \in \mathbb{R}^+$, and $\boldsymbol{A}_1 \in \mathbb{Z}_q^{m\times n}, \boldsymbol{A}_2 \in \mathbb{Z}_q^{m\times k}$. Let $B \in \mathbb{R}$ with $B > \sigma\sqrt{m/2\pi}$. For a value $\boldsymbol{v} \in \mathcal{M}$, the probability that the commitment algorithm defined in Figure 1 with input $\boldsymbol{v}$ generates $\boldsymbol{r}$ and a commitment $\boldsymbol{c}$ such that $(\boldsymbol{c}, \boldsymbol{v}, \boldsymbol{r})$ gets accepted by the unveil algorithm defined in Figure 1 is bounded by*

$$\Pr[\mathsf{Unv}(\mathsf{com}(v,r,e),v,r) = v : r \xleftarrow{\$} \mathbb{Z}_q^n, e \xleftarrow{\$} D_\sigma^m] > 1 - (\frac{B}{\sigma\sqrt{m}}\sqrt{2\pi e}\exp(-\pi(\frac{B}{\sigma\sqrt{m}})^2))^m.$$

Proof: Follows directly from Lemma 2.10 in Micciancio and Regev [19]. Using Lemma 2.10 Micciancio and Regev [19], we can bound the probability that a valid commitment of a message $\boldsymbol{v} \in \mathbb{Z}_q^n$ gets accepted by

$$\Pr(\|\boldsymbol{e}\| \leq B \mid \boldsymbol{e} \xleftarrow{\$} D_\sigma^m) > 1 - \left(\frac{B}{\sigma\sqrt{m}}\sqrt{2\pi e}\exp\left(-\pi(\frac{B}{\sigma\sqrt{m}})^2\right)\right)^m.$$

■

In Section 1 we established that in order to guarantee long-term security of a commitment scheme it is necessary to have an unconditional hiding property. We show in the following that with the correct instantiation LPCom possesses this very property.

For this, we need the following two auxiliary lemmas about the so called smoothing parameter [19]. Roughly speaking, the smoothing parameter bounds the necessary size of a Gaussian error one has to add to a random lattice vectors to completely hide the lattice.

Lemma 1 (Micciancio and Regev [19]) *For any m-dimensional lattice Λ and $\epsilon > 0$,*

$$\eta_\epsilon(\Lambda) \leq \sqrt{\frac{\ln(2m(1+1/\epsilon))}{\pi}} \cdot \lambda_m(\Lambda).$$

Lemma 2 (Gentry, Peikert, and Vaikuntanathan [13]) *Let Λ' be an m-dimensional integer lattice. Then for any $\epsilon \in (0, 1/2)$, any $\sigma \geq \eta_\epsilon(\Lambda')$, the distribution of $(D_\sigma \mod \Lambda')$ is within statistical distance at most 2ϵ of uniform over $(\mathbb{Z}^m \mod \Lambda')$.*

In the following theorem we provide conditions under which LPCom is statistically hiding.

Theorem 2 (hiding) LPCom *is statistically hiding if $\sigma > \sqrt{\frac{\ln(2m(1+1/\epsilon))}{\pi}} \cdot \lambda_m(\Lambda_q(\boldsymbol{A}_2))$ for some ε negligible in κ.*

Proof: Since $\mathbb{Z}_q^m$ is a group, it suffices to show that for $\boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_q^k$ and $\boldsymbol{e} \xleftarrow{\$} D_\sigma^m$, the distribution of $\boldsymbol{A}_2\boldsymbol{r} + \boldsymbol{e} \mod q$ is statistically close to uniform. For $\boldsymbol{x} \in \mathbb{Z}_q^m$ we define

$$p_{\boldsymbol{A}_2}(\boldsymbol{x}) := \Pr[\boldsymbol{A}_2\boldsymbol{r} + \boldsymbol{e} = \boldsymbol{x} \mod q : \boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_q^k, \boldsymbol{e} \xleftarrow{\$} D_\sigma^m].$$

Note that

$$\begin{aligned} p_{\boldsymbol{A}_2}(\boldsymbol{x}) &\sim \sum_{\boldsymbol{r}\in\mathbb{Z}_q^k} \Pr[\boldsymbol{A}_2\boldsymbol{r}+\boldsymbol{e}=\boldsymbol{x} \mod q : \boldsymbol{e} \xleftarrow{\$} D_\sigma^m] \\ &\sim \sum_{\boldsymbol{r}\in\mathbb{Z}_q^k}\sum_{\boldsymbol{v}\in\mathbb{Z}^m} \Pr[\boldsymbol{A}_2\boldsymbol{r}+\boldsymbol{e}+q\boldsymbol{v}=\boldsymbol{x} : \boldsymbol{e} \xleftarrow{\$} D_\sigma^m]. \end{aligned}$$

By definition we have $\Lambda_q(\boldsymbol{A}_2) = \{\boldsymbol{A}_2\boldsymbol{r} \mid \boldsymbol{r}\in\mathbb{Z}^k\} + q\mathbb{Z}^m$. In order to make this representation as a sum unique, we can write $\Lambda_q(\boldsymbol{A}_2) = \{\boldsymbol{A}_2\boldsymbol{r} \mid \boldsymbol{r}\in\{0,1,\ldots,q-1\}^k\}+q\mathbb{Z}^m$, since $\boldsymbol{A}$ has trivial kernel and thus the columns of $\boldsymbol{A}_2$ are $\mathbb{Z}_q$-linearly independent. Consequently, summing over all elements in $\{\boldsymbol{A}_2\boldsymbol{r}+q\boldsymbol{w}_2 \mid \boldsymbol{r}\in\mathbb{Z}_q^k, \boldsymbol{w}_2\in\mathbb{Z}^m\}$ is equivalent to iterating all lattice vectors in $\Lambda_q(\boldsymbol{A}_2)$. This leads to

$$\begin{aligned} p_{\boldsymbol{A}_2}(\boldsymbol{x}) &\sim \sum_{w\in\Lambda_q(\boldsymbol{A}_2)} \Pr[\boldsymbol{w}+\boldsymbol{e}=\boldsymbol{x} : \boldsymbol{e} \xleftarrow{\$} D_\sigma^m] \\ &= \Pr[\boldsymbol{e}=\boldsymbol{x} \mod \Lambda_q(\boldsymbol{A}_2) : \boldsymbol{e} \xleftarrow{\$} D_\sigma^m]. \end{aligned}$$

The constant of proportionality can be determined via

$$\begin{aligned} 1 &= c\sum_{x\in\mathbb{Z}_q^m} \Pr[\boldsymbol{e}=\boldsymbol{x} \mod \Lambda_q(\boldsymbol{A}_2) : \boldsymbol{e} \xleftarrow{\$} D_\sigma^m] \\ &= c\cdot\frac{q^m}{\det(\Lambda_q(\boldsymbol{A}_2))} \sum_{x\in P(\Lambda_q(\boldsymbol{A}_2))} \Pr[\boldsymbol{e}=\boldsymbol{x} \mod \Lambda_q(\boldsymbol{A}_2) : \boldsymbol{e} \xleftarrow{\$} D_\sigma^m] \\ &= c\cdot\frac{q^m}{\det(\Lambda_q(\boldsymbol{A}_2))}, \end{aligned}$$

where $P(\Lambda_q(\boldsymbol{A}_2))$ denotes the fundamental parallelepiped of an arbitrary but fixed basis of $\Lambda_q(\boldsymbol{A}_2)$. This shows that

$$p_{\boldsymbol{A}_2}(\boldsymbol{x}) = \frac{\det(\Lambda_q(\boldsymbol{A}_2))}{q^m}\Pr[D_\sigma^m = \boldsymbol{x} \mod \Lambda_q(\boldsymbol{A}_2)].$$

Since

$$\sigma > \sqrt{\frac{\ln(2m(1+1/\epsilon))}{\pi}}\cdot\lambda_m(\Lambda_q(\boldsymbol{A}_2)),$$

Lemma 1 shows that $\sigma \geq \eta_\epsilon(\Lambda_q(\boldsymbol{A}_2))$. Starting from Lemma 2 with $\Lambda' = \Lambda_q(\boldsymbol{A}_2)$, we can conclude that

$$\begin{aligned} 2\varepsilon &> \mathrm{Dist}(D_\sigma^m \mod \Lambda', \mathrm{Unif}(\mathbb{Z}_q^m) \mod \Lambda') \\ &= \sum_{\boldsymbol{x}\in P(\Lambda')} \left|\Pr[\boldsymbol{e}=\boldsymbol{x} \mod \Lambda' : \boldsymbol{e} \xleftarrow{\$} D_\sigma^m] - \Pr[\boldsymbol{e}=\boldsymbol{x} \mod \Lambda' : \boldsymbol{e} \xleftarrow{\$} \mathbb{Z}_q^m]\right| \\ &= \sum_{\boldsymbol{x}\in P(\Lambda')} \left|\Pr[\boldsymbol{e}=\boldsymbol{x} \mod \Lambda' : \boldsymbol{e} \xleftarrow{\$} D_\sigma^m] - \frac{1}{\det(\Lambda')}\right|. \end{aligned}$$

Therefore, the statistical distance between the distribution Y of $\boldsymbol{A}_2\boldsymbol{r}+\boldsymbol{e} \mod q$, where $\boldsymbol{r} \xleftarrow{\$} \mathbb{Z}_q^k, \boldsymbol{e} \xleftarrow{\$} D_\sigma^m$ and the uniform distribution in $\mathbb{Z}_q$ is bounded by

$$
\begin{aligned}
&2\,\mathrm{Dist}(Y, \mathrm{Unif}(\mathbb{Z}_q^m)) \\
&\quad= \sum_{\boldsymbol{x}\in\mathbb{Z}_q^m} \left|\Pr[\boldsymbol{x}=\boldsymbol{y} : \boldsymbol{y} \xleftarrow{\$} Y] - \Pr[\boldsymbol{x}=\boldsymbol{y} : \boldsymbol{y} \xleftarrow{\$} \mathbb{Z}_q^m]\right| \\
&\quad= \frac{q^m}{\det(\Lambda')} \sum_{\boldsymbol{x}\in P(\Lambda')} \left|\Pr[\boldsymbol{x}=\boldsymbol{y} : \boldsymbol{y} \xleftarrow{\$} Y] - q^{-m}\right| \\
&\quad= \frac{q^m}{\det(\Lambda')} \sum_{\boldsymbol{x}\in P(\Lambda')} \left|\frac{\det(\Lambda')}{q^m} \Pr[\boldsymbol{x}=\boldsymbol{y} \mod \Lambda' : \boldsymbol{y} \xleftarrow{\$} D_\sigma^m] - q^{-m}\right| \\
&\quad= \sum_{\boldsymbol{x}\in P(\Lambda')} \left|\Pr[\boldsymbol{x}=\boldsymbol{y} \mod \Lambda' : \boldsymbol{y} \xleftarrow{\$} D_\sigma^m] - \frac{1}{\det(\Lambda')}\right| \leq 2\varepsilon,
\end{aligned}
$$

which shows that $\mathrm{Dist}(Y, \mathrm{Unif}(\mathbb{Z}_q^m)) \leq \varepsilon$. ■

Since LPCom with the correct instantiation is statistically hiding it can only achieve computational bindingness. We show that LPCom is computationally binding by relating its binding property to solving the hard lattice problem α-SVP. In Section 2.2 we already discussed the hardness of this problem.

Theorem 3 (binding) LPCom *is computationally binding as long as for any probabilistic polynomial-time algorithm the probability of solving α-SVP in the m-dimensional q-ary lattice $\Lambda_q(\boldsymbol{A})$ with approximation factor $\alpha = \frac{2B}{\lambda_1(\Lambda_q(\boldsymbol{A}))}$ is negligible in the security parameter k.*

Proof: Assume to the contrary that one can find two different opening triples that lead to the same (valid) commitment in polynomial time, i.e. one can find distinct $(\boldsymbol{v}_1, \boldsymbol{r}_1, \boldsymbol{e}_1), (\boldsymbol{v}_2, \boldsymbol{r}_2, \boldsymbol{e}_2) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^k \times \mathbb{Z}_q^m$ with $\|\boldsymbol{e}_1\| \leq B, \|\boldsymbol{e}_2\| \leq B$ such that $\boldsymbol{A}_1\boldsymbol{v}_1 + \boldsymbol{A}_2\boldsymbol{r}_1 + \boldsymbol{e}_1 = \boldsymbol{A}_1\boldsymbol{v}_2 + \boldsymbol{A}_2\boldsymbol{r}_2 + \boldsymbol{e}_2$. We can rewrite this equation as

$$
\boldsymbol{A}\begin{pmatrix}\boldsymbol{v}_1 - \boldsymbol{v}_2 \\ \boldsymbol{r}_1 - \boldsymbol{r}_2\end{pmatrix} = \boldsymbol{A}_1(\boldsymbol{v}_1 - \boldsymbol{v}_2) + \boldsymbol{A}_2(\boldsymbol{r}_1 - \boldsymbol{r}_2) = \boldsymbol{e}_2 - \boldsymbol{e}_1.
$$

Since A has trivial kernel and the triples $(\boldsymbol{v}_1, \boldsymbol{r}_1, \boldsymbol{e}_1)$ and $(\boldsymbol{v}_2, \boldsymbol{r}_2, \boldsymbol{e}_2)$ are distinct, we have $\boldsymbol{e}_2 - \boldsymbol{e}_1 \neq \boldsymbol{0}$. This means we found a non-zero vector $\boldsymbol{e} = \boldsymbol{e}_2 - \boldsymbol{e}_1$ in the lattice $\Lambda_q(\boldsymbol{A})$ of length at most $2B$ in polynomial time, a contradiction. ■

3.3 Instantiation

While LPCom allows for many different instantiations that offer trade-offs between security, commitment size, secret sizes, and efficiency, it is useful to have a specific parameter set in mind. In the case of LPCom this is particularly helpful since the parameter selection such that LPCom is statistically hiding and

computationally binding is non-trivial. Theorem 4 shows that the parameter set proposed in Figure 2 leads to a secure and correct instantiation. The remaining parameter to tweak the security is k.

Recall that breaking the binding property of LPCom requires an attacker to find a non-zero lattice of length at most $B = \sqrt{m}\sigma$. A straightforward calculation shows that with the instantiation given in Figure 2 such a vector would have hermite factor $\delta = k^{5/(36k)}$. This value tends to one very fast, which implies that finding such a vector is considered computationally infeasible even for moderate values of k. A possible instantiation would be to choose $k = 128$, which leads to a hermite factor of approximately $\delta = 1.005$. Lindner and Peikert [15] estimated the time to find such a vector to be more than 2^{128} seconds, which is way beyond what is possible for the foreseeable future. Note, however, that the parameter sets presented in Figure 2 are not at all optimized for concrete instantiations. Since this work is mainly dealing with long-term security, we leave the task to find optimized parameter sets that meet specific security levels as an interesting future work.

Parameter	Meaning	Instantiation	
		Value	Assymptotic
κ	statistical distance parameter	100	constant
k	security parameter		$\mathcal{O}(k)$
n	message space rank	k	$\mathcal{O}(k)$
m	lattice dimension	$3k$	$\mathcal{O}(k)$
q	modulus	k^2	$\mathcal{O}(k^2)$
σ	error size	$k^{5/4}$	$\mathcal{O}(k^{5/4})$
B	error bound	$\sqrt{m}\sigma$	$\mathcal{O}(k^{7/4})$

Fig. 2. Parameter Set for LPCom

Theorem 4 *Consider* LPCom. *Suppose that Equation (1) holds for the lattice* $\Lambda_q(\boldsymbol{A})$*, and that for any PPT algorithm the probability of solving* α*-SVP in* $\Lambda_q(\boldsymbol{A})$ *with (polynomial) approximation factor* $\alpha \leq 2\sqrt{3} \cdot k^{7/4}$ *is negligible in the security parameter* k*.*
Then for sufficiently large security parameter k*,* LPCom *with the instantiation from Figure 2 is correct, statistically hiding, and computationally binding.*

Proof:

Correctness: Following Theorem 1, the probability that a valid commitment gets rejected is bounded from above by

$$\left(\frac{B}{\sigma\sqrt{m}}\sqrt{2\pi e}\exp(-\pi(\frac{B}{\sigma\sqrt{m}})^2)\right)^m = \left(\sqrt{2\pi e}\exp(-\pi)\right)^m,$$

which is smaller than $2^{-\kappa}$ for sufficiently large $m = 3k$.

Hiding: For $\varepsilon = 2^{-\kappa}$ and m big enough, Equation (1) leads to

$$\sqrt{\frac{\ln(2m(1+1/\epsilon))}{\pi}} \cdot \lambda_m(\Lambda_q(\boldsymbol{A})) \leq \sqrt{\frac{\ln(2m(1+2^\kappa))}{\pi}} \cdot \sqrt{\frac{m}{2\pi}} \det(\Lambda)^{1/m}$$
$$\leq \ln(m) \cdot \sqrt{m} \cdot q^{1/3}.$$

Inserting the parameters from Figure 2 shows that

$$\sqrt{\frac{\ln(2m(1+1/\epsilon))}{\pi}} \cdot \lambda_m(\Lambda_q(\boldsymbol{A})) \leq \ln(3k) \cdot \sqrt{3k} \cdot k^{2/3} = \sqrt{3}\ln(3k) \cdot k^{7/6} < k^{5/4} = \sigma$$

for k sufficiently large. The rest follows from Theorem 2.

Binding: By Theorem 3 it suffices to show the upper bound on $\alpha = \frac{2B}{\lambda_1(\Lambda_q(\boldsymbol{A}))}$ stated in the theorem. Since trivially $\lambda_1(\Lambda_q(\boldsymbol{A})) \geq 1$ holds, we have

$$\alpha = \frac{2\sqrt{m}\sigma}{\lambda_1(\Lambda_q(\boldsymbol{A}))} \leq 2\sqrt{m}\sigma = 2\sqrt{3} \cdot k^{7/4}.$$

■

4 Proof of Knowledge for LPCom

In a proof of knowledge, a prover P convinces a verifier V that it has knowledge of a secret without revealing anything about the secret apart from what is revealed by the claim itself (see Bellare and Goldreich [2] for a formal definition). In our case the claim is a commitment and the secret consists of the corresponding opening values.

Fist, in Section 4.1 we give a definition of Σ^*-protocols. Then in Section 4.2 the proof of knowledge protocol for LPCom is defined. Finally, Section 4.3 provides an instatiation of LPCom such that the proof of knowledge protocol is a Σ^*-protocol, while still being statistically hiding and computationally binding.

4.1 Σ^*-Protocols

Proofs of knowledge are usually performed using Σ-protocols. Formal definitions of Σ-protocols have been provided, for instance, by Cramer [7] or Damgård [9]. In [3, 4], Benhamouda et al. introduced a modification of Σ-protocols called Σ'-protocols and discussed their relation to Σ-protocols. However, there are two slight differences between the Σ'-protocols used by Benhamouda et al. and our protocols. The first is due to the fact that LPCom is statistically hiding, therefore we are able to achieve a statistical zero-knowledge property instead of just the computational one of Σ'-protocols. The second is that in consequence LPCom can only be computationally binding, hence we have to relax the special soundness condition of Σ'-protocols to *computationally special soundness* (adapted from Ambainis et al. [1] and Pass [21]). This leads to the following modified definition.

Definition 1 *Let* (P, V) *be a two-party protocol, where* P *and* V *are PPT, and let* $\mathcal{L}, \mathcal{L}' \subseteq \{0,1\}^*$ *be languages with witness relations* $\mathcal{R} \subseteq \mathcal{R}' \subseteq \{0,1\}^* \times \{0,1\}^*$. *(That is, for all* $a \in \{0,1\}^*$ *we have* $a \in \mathcal{L}^{(\prime)}$ *if and only if there exists some* $b \in \{0,1\}^*$ *such that* $(a,b) \in \mathcal{R}^{(\prime)}$. *In this case* b *is called a witness for* $a \in \mathcal{L}^{(\prime)}$.*) Then* (P, V) *is called a* Σ^**-protocol for* $\mathcal{R}, \mathcal{R}'$ *with completeness error* α, *challenge set* $\mathcal{C} = \{0,1\}$, *public input* x, *and private input* w, *if and only if it satisfies the following conditions:*

Three-move form: *The protocol is of the following form: The prover* P, *on input* (x, w), *computes a commitment* t *and sends it to* V. *The verifier* V, *on input* x, *then draws a challenge* $c \xleftarrow{\$} \mathcal{C}$ *and sends it to* P. *The prover sends a response* s *to the verifier. Depending on the public input* x *and the protocol transcript* $(\mathsf{t}, \mathsf{c}, \mathsf{s})$, *the verifier finally accepts (*$ok = 1$*) or rejects (*$ok = 0$*) the proof. The protocol transcript* $(\mathsf{t}, \mathsf{c}, \mathsf{s})$ *is called accepting, if the verifier accepts the protocol run.*

Completeness: *Whenever* $(x, w) \in \mathcal{R}$, *the verifier* V *accepts with probability at least* $1 - \alpha$.

Computational special soundness: *There exists a PPT algorithm* E *(the knowledge extractor) such that for any PPT algorithm* A *(the adversary), we have that*

$$\begin{aligned}\Pr[(x,w) \notin \mathcal{R}') \wedge (ok' = ok'' = 1) :\\ (x,(t,0,s'),(t,1,s'')) \leftarrow \mathsf{A},\\ ok' \leftarrow \mathsf{V}(x,(t,0,s')), ok'' \leftarrow \mathsf{V}(x,(t,1,s'')),\\ w \leftarrow \mathsf{E}(x,(t,0,s'),(t,1,s''))]\end{aligned}$$

is negligible.

Special honest-verifier statistical zero-knowledge (HVSZK): *There exists a PPT algorithm* S *(the simulator) taking* $x \in \mathcal{L}$ *and* $\mathsf{c} \in \mathcal{C}$ *as inputs, that outputs* (t, s) *so that the triple* $(\mathsf{t}, \mathsf{c}, \mathsf{s})$ *is statistically indistinguishable from an accepting protocol transcript generated by a real protocol run.*

Remark 1 *We give a few informal remarks on the properties defined in Definition 1.*

1. *The completeness error* α *accounts for the fact that even an honest prover is not able to answer all challenges correctly. Note that the same holds true for the* Σ' *protocol defined by Benhamouda et al.*
2. *The intuitive meaning of computational special soundness is the following. If a public input* x *and two accepting transcripts that answer both challenges to the same commitment are given as an output of a PPT algorithm, one can extract a witness* w *for* $x \in \mathcal{L}'$ *in polynomial time (with overwhelming probability).*
3. *In the previous remark we mentioned that* w *is a witness for* $x \in \mathcal{L}'$ *whereas an honest prover is supposed to know a witness for* $x \in \mathcal{L}$. *However, this is not a problem as long as finding witnesses for* $x \in \mathcal{L}'$ *is still hard, as it will be in our case.*

4. *Intuitively, the zero-knowledge property means that real protocol runs with honest prover and verifier can efficiently be simulated, hence revealing no additional information about the witness.*

4.2 Proof of Knowledge Protocol

One well known approach for proving knowledge in perfect zero-knowledge fashion is to use a cut-and-choose based proof similar to the one proposed by Schnorr [23]. Our approach is based on a modification of this proposal introduced by Benhamouda et al. [3, 4].

Informally, the main idea of the protocol is the following. The prover wants to show that it can open a commitment $\boldsymbol{c} = \mathsf{com}(\boldsymbol{v}, \boldsymbol{r}, \boldsymbol{e})$. In order to do so it chooses a random commitment $\boldsymbol{c}' = \mathsf{com}(\boldsymbol{v}', \boldsymbol{r}', \boldsymbol{e}')$ and sends it to the verifier. The prover is then asked by the verifier to either open $\boldsymbol{c}'$ or $\boldsymbol{c}+\boldsymbol{c}'$. Note that the prover can indeed open $\boldsymbol{c}+\boldsymbol{c}'$ (with overwhelming probability), since $\boldsymbol{c}+\boldsymbol{c}' = \mathsf{com}(\boldsymbol{v}+\boldsymbol{v}', \boldsymbol{r}+\boldsymbol{r}', \boldsymbol{e}+\boldsymbol{e}')$. The hidden caveat of this approach is that by opening $\boldsymbol{c}+\boldsymbol{c}'$ the verifier learns $\boldsymbol{e}+\boldsymbol{e}'$, which is Gaussian distributed centered around $\boldsymbol{e}$ and hence contains information about $\boldsymbol{e}$, and thus about $\boldsymbol{v}$. This leakage of information would violate the hiding property of the commitment scheme. In order to overcome this problem one uses a technique called rejection sampling, described in the following.

Rejection sampling is a standard technique used in statistics and well known in lattice-based cryptography (see [16, 17]). It is used to transform one probability distribution into another. This is needed for our protocol to transform a Gaussian distribution that is centered around some $\boldsymbol{e}$ into one that is centered around zero and independent of $\boldsymbol{e}$.

Theorem 5 *Let V be a subset of $\mathbb{Z}^m$ in which all elements have norm at most T, $\alpha \in \mathbb{R}^+$ and h be a probability distribution on V. For $M = \exp(\frac{2\sqrt{\kappa}}{\alpha} + \frac{1}{2\alpha^2})$ and $\sigma \leq \alpha T$, the statistical distance of the output distribution of*

1. $\boldsymbol{v} \xleftarrow{\$} h$
2. $\boldsymbol{z} \xleftarrow{\$} D^m_{\boldsymbol{v},\sigma}$
3. *output $(\boldsymbol{v}, \boldsymbol{z})$ with probability $\min\left(\frac{D^m_\sigma(\boldsymbol{z})}{M D^m_{\boldsymbol{v},\sigma}(\boldsymbol{z})}, 1\right)$, else output $\perp$*

and the output distribution of

1. $\boldsymbol{v} \xleftarrow{\$} h$
2. $\boldsymbol{z} \xleftarrow{\$} D^m_\sigma$
3. *output $(\boldsymbol{v}, \boldsymbol{z})$ with probability $1/M$, else output $\perp$*

is bounded by $\exp(-\kappa)$.

Proof: Similar to the proof of Theorem 4.6 by Lyubashevsky [17]. ■

The Protocol

Let $n, m, \kappa, k, q \in \mathbb{Z}$, $\sigma, \sigma' \in \mathbb{R}^+$, $T = \sqrt{m}\sigma, \alpha = \sigma'/T, M = \exp(\frac{2\sqrt{\kappa}}{\alpha} + \frac{1}{2\alpha^2})$, $\boldsymbol{A}_1 \in \mathbb{Z}_q^{m \times n}, \boldsymbol{A}_2 \in \mathbb{Z}_q^{m \times k}$. Also let $\mathcal{M} \subset \mathbb{Z}_q^n$ be a subgroup and

$$\mathsf{com} : \mathcal{M} \times \mathbb{Z}_q^k \times \mathbb{Z}_q^m \to \mathbb{Z}_q^m, (\boldsymbol{v}, \boldsymbol{r}, \boldsymbol{e}) \mapsto \boldsymbol{A}_1\boldsymbol{v} + \boldsymbol{A}_2\boldsymbol{r} + \boldsymbol{e}$$

be the corresponding commitment function described in Figure 1. Furthermore, let $B \in \mathbb{R}^+$ and Unv and Unv_{2B} be the corresponding unveil functions with bound B and $2B$ respectively as described in Figure 1.
Finally, let Com_a be an auxiliary computationally binding and statistically hiding commitment scheme with message space $\mathbb{Z}_q^m$ and Unv_a be the corresponding unveil function.
Figure 3 describes the proof of knowledge protocol for LPCom.

The public input is $\boldsymbol{c} \in \mathbb{Z}_q^m$ and the private input is $(\boldsymbol{v}, \boldsymbol{r}) \in \mathcal{M} \times \mathbb{Z}_q^k$.

1. P computes $\boldsymbol{e} = \boldsymbol{c} - \boldsymbol{A}_1\boldsymbol{v} - \boldsymbol{A}_2\boldsymbol{r}$. Then P secretly samples $\boldsymbol{v}' \xleftarrow{\$} \mathcal{M}$, $\boldsymbol{r}' \xleftarrow{\$} \mathbb{Z}_q^k$, $\boldsymbol{e}' \xleftarrow{\$} D_{\sigma'}^m$ and computes the commitment $\boldsymbol{c}' = \mathsf{com}(\boldsymbol{v}', \boldsymbol{r}', \boldsymbol{e}')$. P then computes an auxiliary commitment $\boldsymbol{c}'_a$ and corresponding opening randomness value $\boldsymbol{r}'_a$ using Com_a with input $\boldsymbol{c}'$ and sends $\boldsymbol{c}'_a$ to V.
2. V sends a random bit $b \in \{0, 1\}$ to P.
3. P computes $\boldsymbol{v}'_b = \boldsymbol{v}' + b\boldsymbol{v}$ and $\boldsymbol{r}'_b = \boldsymbol{r}' + b\boldsymbol{r}$.
   if $b = 0$:
   P sends $(\boldsymbol{c}', \boldsymbol{r}'_a, \boldsymbol{v}'_b, \boldsymbol{r}'_b)$ to V.
   if $b = 1$:
   With probability
   $$p = \min\left(\frac{D_{\sigma'}^m(\boldsymbol{e}')}{M D_{\boldsymbol{e},\sigma'}^m(\boldsymbol{e}')}, 1\right),$$
   P sends $(\boldsymbol{c}', \boldsymbol{r}'_a, \boldsymbol{v}'_b, \boldsymbol{r}'_b)$ to V, and $\perp$ otherwise.
4. V accepts iff P did not send $\perp$ and $\mathsf{Unv}_a(\boldsymbol{c}'_a, \boldsymbol{c}', \boldsymbol{r}'_a) = \boldsymbol{c}'$ and $\boldsymbol{v}'_b \in \mathcal{M}$ and $\mathsf{Unv}(\boldsymbol{c}' + b\boldsymbol{c}, \boldsymbol{v}'_b, \boldsymbol{r}'_b) = \boldsymbol{v}'_b$.

Fig. 3. Proof of Knowledge

Remark 2 *The auxiliary commitment used during the protocol is needed to guarantee the zero-knowledge property in case of an abort happening. This can be seen in the proof of Theorem 7 in the following Section 4.3.*

4.3 Instantiation

In Figure 4 we propose a set of parameters that supports the proof of knowledge protocol (see Theorem 7) while still preserving the statistically hiding and computationally binding properties of LPCom, as shown in the following theorem.

Param.	Meaning	Instantiation	
		Value	Assymptotic
κ	statistical distance parameter	100	constant
k	security parameter		$\mathcal{O}(k)$
n	message space rank		constant
m	lattice dimension	$3k$	$\mathcal{O}(k)$
q	modulus	k^3	$\mathcal{O}(k^3)$
σ	error size	$k^{1.75}$	$\mathcal{O}(k^{1.75})$
σ'	error size in protocol	$4\sqrt{\kappa m}\sigma$	$\mathcal{O}(k^{2.25})$
B	error bound	$\sqrt{m}\sigma'$	$\mathcal{O}(k^{2.75})$
$\frac{1}{M}$	answering probability	$e^{-\frac{1}{2}-\frac{1}{2\kappa}}$	$\frac{1}{\sqrt{e^{1.01}}}$

Fig. 4. Parameter Set for Proof of Knowledge

Theorem 6 *Consider* LPCom. *Suppose that Equation (1) holds for the lattice* $\Lambda_q(\boldsymbol{A})$*, and that for any PPT algorithm the probability of solving* α*-SVP in* $\Lambda_q(\boldsymbol{A})$ *with (polynomial) approximation factor* $\alpha \leq 2\sqrt{3}\cdot k^{7/4}$ *is negligible in the security parameter* k*.*
Then for sufficiently large security parameter k*,* LPCom *with the instantiation from Figure 4 is correct, statistically hiding, and computationally binding.*

Proof: Similar to the proof of Theorem 4. ■

The following theorem shows that the protocol defined in Figure 3 is in fact a Σ^*-protocol for the instantiation proposed in Figure 4.

Theorem 7 *With the parameters as given in Figure 4 and sufficiently large security parameter* k *the protocol defined in Figure 3 is a* Σ^**-protocol with completeness error overwhelmingly close to* $\frac{1}{2}-\frac{1}{2M}$ *for the relations*

$$\mathcal{R} = \{(\boldsymbol{c},(\boldsymbol{v},\boldsymbol{r})) \in \mathbb{Z}_q^m \times (\mathcal{M}\times\mathbb{Z}_q^k) \mid \boldsymbol{v} = \mathsf{Unv}(\boldsymbol{c},\boldsymbol{v},\boldsymbol{r})\} \text{ and}$$
$$\mathcal{R}' = \{(\boldsymbol{c},(\boldsymbol{v},\boldsymbol{r})) \in \mathbb{Z}_q^m \times (\mathcal{M}\times\mathbb{Z}_q^k) \mid \boldsymbol{v} = \mathsf{Unv}_{2B}(\boldsymbol{c},\boldsymbol{v},\boldsymbol{r})\}.$$

Proof: Obviously, the protocol is given in the three-move form.
Completeness: Assume P is an honest prover and let $\boldsymbol{c}$ be the public input and $(\boldsymbol{v},\boldsymbol{r})$ with $(\boldsymbol{c},(\boldsymbol{v},\boldsymbol{r})) \in \mathcal{R}$ be the private input of P. First consider the case $b=0$. In this case P always answers the challenge by sending $(\boldsymbol{c}',\boldsymbol{r}'_a,\boldsymbol{v}',\boldsymbol{r}')$. Note that $\boldsymbol{v}' \in \mathcal{M}$. We have $\mathsf{Unv}_a(\boldsymbol{c}'_a,\boldsymbol{v}'_a,\boldsymbol{r}'_a) = \boldsymbol{v}'_a$ (with overwhelming probability), so the first of the three accepting conditions is satisfied. Since $\boldsymbol{c}' = \mathsf{com}(\boldsymbol{v}',\boldsymbol{r}',\boldsymbol{e}')$ and $\boldsymbol{e}' \xleftarrow{\$} D_\sigma^m$, we have $\mathsf{Unv}(\boldsymbol{c}',\boldsymbol{v}',\boldsymbol{r}') = \boldsymbol{v}'$ with overwhelming probability by Theorem 1.
Now consider the case $b=1$. Then by Theorem 5 the prover P answers the challenge by sending $(\boldsymbol{c}',\boldsymbol{r}'_a,\boldsymbol{v}'+\boldsymbol{v},\boldsymbol{r}'+\boldsymbol{r})$ with probability overwhelmingly close to $\frac{1}{M}$, and $\perp$ otherwise. In the following we consider the case that P does not abort. Note that $\boldsymbol{v}'+\boldsymbol{v} \in \mathcal{M}$. As before, $\mathsf{Unv}_a(\boldsymbol{c}'_a,\boldsymbol{v}'_a,\boldsymbol{r}'_a) = \boldsymbol{v}'_a$ (with overwhelming probability), thus the first of the three accepting conditions is satisfied. Notice

that, after the rejection sampling performed in the protocol, the distribution of $\boldsymbol{e}' + \boldsymbol{e}$ is statistically close to $D_{\sigma'}^m$ by Theorem 5. Hence it follows from Theorem 1 that $\mathsf{Unv}(\boldsymbol{c}' + \boldsymbol{c}, \boldsymbol{v}' + \boldsymbol{v}, \boldsymbol{r}' + \boldsymbol{r}) = \boldsymbol{v}' + \boldsymbol{v}$ with overwhelming probability, since $\boldsymbol{c}' + \boldsymbol{c} = \mathsf{com}(\boldsymbol{v}' + \boldsymbol{v}, \boldsymbol{r}' + \boldsymbol{r}, \boldsymbol{e}' + \boldsymbol{e})$.
In conclusion the verifier V accepts with probability overwhelmingly close to $\frac{1}{2}(1 + \frac{1}{M})$, since both cases $b = 0$ and $b = 1$ are equally likely.
Computational special soundness: Assume a public input $\boldsymbol{c}$ and two accepting transcripts $(\boldsymbol{c}'_a, 0, (\boldsymbol{c}', \boldsymbol{r}'_a, \boldsymbol{v}'_0, \boldsymbol{r}'_0))$ and $(\boldsymbol{c}'_a, 1, (\boldsymbol{c}'', \boldsymbol{r}''_a, \boldsymbol{v}''_1, \boldsymbol{r}''_1))$ are given as an output of a probabilistic polynomial time algorithm A. Since the auxiliary commitment scheme Com_a is computationally binding, we have $\boldsymbol{c}' = \boldsymbol{c}''$ with overwhelming probability, which we will therefore assume from now on. Thus by definition of the accepting condition of the protocol we have $\boldsymbol{v}_0, \boldsymbol{v}_1 \in \mathcal{M}$ and

$$\|\boldsymbol{c}' - \boldsymbol{A}_1\boldsymbol{v}'_0 - \boldsymbol{A}_2\boldsymbol{r}'_0\| \leq B \text{ and } \|\boldsymbol{c}' + \boldsymbol{c} - \boldsymbol{A}_1\boldsymbol{v}''_1 - \boldsymbol{A}_2\boldsymbol{r}''_1\| \leq B.$$

Hence $\boldsymbol{v}''_1 - \boldsymbol{v}'_0 \in \mathcal{M}$ and by linearity and the triangular inequation we obtain

$$\|\boldsymbol{c} - \boldsymbol{A}_1(\boldsymbol{v}''_1 - \boldsymbol{v}'_0) - \boldsymbol{A}_2(\boldsymbol{r}''_1 - \boldsymbol{r}'_0)\| \leq 2B,$$

and thus $(c, (\boldsymbol{v}''_1 - \boldsymbol{v}'_0, \boldsymbol{r}''_1 - \boldsymbol{r}'_0)) \in \mathcal{R}'$.
HVSZK: Let $\boldsymbol{c}$ and $b \in \{0, 1\}$ be given. The simulator S samples $\boldsymbol{v}' \xleftarrow{\$} \mathcal{M}$, $\boldsymbol{r}' \xleftarrow{\$} \mathbb{Z}_q^k$, $\boldsymbol{e}' \xleftarrow{\$} D_{\sigma'}^m$ and computes the commitment $\boldsymbol{c}' = \mathsf{com}(\boldsymbol{v}', \boldsymbol{r}', \boldsymbol{e}') - b\boldsymbol{c}$. Then S computes an auxiliary commitment $\boldsymbol{c}_a$ and corresponding opening randomness value $\boldsymbol{r}_a$ using Com_a with input $\boldsymbol{c}'$.
First consider the case $b = 0$. In this case S outputs $(\boldsymbol{c}_a, 0, (\boldsymbol{c}', \boldsymbol{r}_a, \boldsymbol{v}', \boldsymbol{r}'))$. Obviously this is statistically indistinguishable from real protocol runs.
Now consider the case $b = 1$. With probability $\frac{1}{M}$ the simulator outputs $(\boldsymbol{c}_a, 1, (\boldsymbol{c}', \boldsymbol{r}_a, \boldsymbol{v}', \boldsymbol{r}'))$. Otherwise, S computes an auxiliary commitment $\widehat{\boldsymbol{c}_a}$ and corresponding opening randomness value $\widehat{\boldsymbol{r}_a}$ using Com_a with input $\mathbf{0}$ and outputs $(\widehat{\boldsymbol{c}_a}, 1, \perp)$. From the statistical hidingness of the auxiliary commitment and Theorem 5, it follows that this output is statistically indistinguishable from real protocol runs. (In order to see this, recall that the rejection sampling performed in the protocol is used to make $\boldsymbol{e} + \boldsymbol{e}'$ Gaussian distributed around zero, which consequently makes $\boldsymbol{e}'$ Gaussian distributed around $-\boldsymbol{e}$.) ■

Remark 3 *Note that finding witnesses for $\mathcal{L}'$ is nearly as hard as finding witnesses for $\mathcal{L}$. In order to see this notice the following. To find witnesses for $x \in \mathcal{L}$ one needs to solve α-SVP in a certain lattice with approximation factor α. This is exactly what the binding property of* LPCom *is based on (see Theorem 3) and addressed in Theorem 6. Finding witnesses for $x \in \mathcal{L}'$ implies solving (2α)-SVP in the same lattice. However, doubling the desired approximation factor does not make α-SVP considerably easier.*

In practice, the probability of a dishonest prover (i.e. one that is not able to open $\boldsymbol{c}$) to convince the verifier that it can open c should be negligible. At the same time, the probability that an honest prover fails to convince the verifier that it can open the commitment should be negligible as well. To achieve this, it

is necessary to repeat the proof of knowledge protocol multiple times in order to amplify the advantage an honest prover has over a dishonest prover. The verifier then must make a decision whether the prover is honest or not, based on the percentage of accepting protocol runs.[6] We now discuss this in further detail.

In N repetitions of the protocol, an honest prover is expected to be able to answer $N/2(1 + 1/M)$ challenges, while a dishonest prover is expected to only give $N/2$ correct answers. A natural choice for the verifier is therefore to accept the prover as honest if and only if it receives more than $N/2(1+1/(2M))$ correct answers. In the following, we will show that $N = 4kM^2$ is a good choice for the number of protocol runs that provides the desired result when using this criterion.

For this, we will model the number of correct answers given by an honest prover as the output of a binomial distribution with parameters N and $p = (1+1/M)/2$. It is well known that for large N, this binomial distribution can be approximated by a continuous Gaussian distribution with mean Np and standard deviation $\sqrt{Np(1-p)}$. Let $p_{\perp,h}$ denote the probability that, given the natural accepting condition described above, the verifier does not accept an honest prover as such after N protocol runs. An easy calculation shows that

$$\begin{aligned} p_{\perp,h} &\approx \Pr[x \leq N(1 + 1/(2M))/2 : x \xleftarrow{\$} \overline{D}_{Np,\sqrt{Np(1-p)}}] \\ &= \Pr[x \leq -N/(4M) : x \xleftarrow{\$} \overline{D}_{\sqrt{N(1-1/M^2)/4}}]. \end{aligned}$$

With $N = 4kM^2 > 4k(M^2 - 1)$, this leads to the estimation

$$\begin{aligned} p_{\perp,h} &\approx \Pr[x \leq -\frac{4k(M^2-1)}{4M} : x \xleftarrow{\$} \overline{D}_{\sqrt{4k(M^2-1)(1-1/M^2)/4}}] \\ &= \Pr[x \leq -\sqrt{k} \cdot \sqrt{k}\frac{(M^2-1)}{M} : x \xleftarrow{\$} \overline{D}_{\sqrt{k}\frac{(M^2-1)}{M}}] \end{aligned}$$

The well-known fact

$$\Pr[x > \sqrt{k}\sigma : x \xleftarrow{\$} \overline{D}_\sigma] \leq \frac{\exp(-\pi k)}{2\pi\sqrt{k}},$$

shows that we can estimate the probability of an honest prover to get rejected to be negligible in k.

The analysis of accepting a dishonest prover is similar. Since it has only a success probability overwhelmingly close to $p = 1/2$, we can estimate its probability $p_{a,d}$ to get accepted by

$$\begin{aligned} p_{a,d} &\approx \Pr[x > N(1 + 1/(2M))/2 : x \xleftarrow{\$} \overline{D}_{Np,\sqrt{Np(1-p)}}] \\ &= \Pr[x > kM : x \xleftarrow{\$} \overline{D}_{\sqrt{k}M}]. \end{aligned}$$

[6] An obvious modification of the protocol is to identify a dishonest prover as soon as it does not provide a valid answer to challenge b=0. However, we do not expect a big improvement, since we have to repeat the protocol often enough to ensure that an honest prover gets accepted.

5 Prolonging the Security of LPCom

This section shows how to prolong the bindingness of our unconditionally hiding commitment scheme LPCom in the following way. Given an old commitment to a certain value the commitment owner generates a new commitment - with *renewed parameters* - to the same value. The owner then proves in perfect zero-knowledge fashion that it can open the old and the new commitment to the same value. This effectively rejuvenates the owner's bind to the value in question, without damaging secrecy in any way, as long as both commitment schemes are unconditionally hiding.

A protocol for the proofs of message equality needed in the prolonging procedure is presented in Section 5.1. Details on the prolonging procedure itself are described in Section 5.2.

5.1 Proof of Message Equality

In a proof of message equality protocol, a prover P convinces a verifier V that it can unveil two commitments generated using two (different) instantiations of LPCom to the same value. As long as the two commitments are computationally binding, the verifier then can be assured that the prover committed to the same value in both commitments. The basic idea of our approach is to run two proof of knowledge protocols in parallel using the same auxiliary value and is based on the methods of [6] and [20].

Let $n, m_1, m_2, k_1, k_2, q_1, q_2 \in \mathbb{Z}$, $\sigma_1, \sigma_2, \sigma_1', \sigma_2' \in \mathbb{R}^+$, $T_1 = \sqrt{m_1}\sigma_1, \alpha_1 = \sigma_1'/T_1$, $M_1 = \exp(\frac{2\sqrt{\kappa}}{\alpha_1} + \frac{1}{2\alpha_1^2})$, $T_2 = \sqrt{m_2}\sigma_2, \alpha_2 = \sigma_2'/T_2$, $M_2 = \exp(\frac{2\sqrt{\kappa}}{\alpha_2} + \frac{1}{2\alpha_2^2})$, $\boldsymbol{A}_{1,1} \in \mathbb{Z}_{q_1}^{m_1 \times n}, \boldsymbol{A}_{1,2} \in \mathbb{Z}_{q_1}^{m_1 \times k_1}$, $\boldsymbol{A}_{2,1} \in \mathbb{Z}_{q_2}^{m_2 \times n}$, $\boldsymbol{A}_{2,2} \in \mathbb{Z}_{q_2}^{m_2 \times k_2}$. Also let $\mathcal{M}_1 \subset \mathbb{Z}_{q_1}^n$ and $\mathcal{M}_2 \subset \mathbb{Z}_{q_2}^n$ be two isomorphic subgroups, $\varphi : \mathcal{M}_1 \to \mathcal{M}_2$ be a group isomorphism, and

$$\mathsf{com}_1 : \mathcal{M}_1 \times \mathbb{Z}_{q_1}^{k_1} \times \mathbb{Z}_{q_1}^{m_1} \to \mathbb{Z}_{q_1}^{m_1}, \qquad (\boldsymbol{v}_1, \boldsymbol{r}_1, \boldsymbol{e}_1) \mapsto \boldsymbol{A}_{1,1}\boldsymbol{v}_1 + \boldsymbol{A}_{1,2}\boldsymbol{r}_1 + \boldsymbol{e}_1$$
$$\mathsf{com}_2 : \mathcal{M}_2 \times \mathbb{Z}_{q_2}^{k_2} \times \mathbb{Z}_{q_2}^{m_2} \to \mathbb{Z}_{q_2}^{m_2}, \qquad (\boldsymbol{v}_2, \boldsymbol{r}_2, \boldsymbol{e}_2) \mapsto \boldsymbol{A}_{2,1}\boldsymbol{v}_2 + \boldsymbol{A}_{2,2}\boldsymbol{r}_2 + \boldsymbol{e}_2$$

be the corresponding respective commitment functions described in Figure 1. We say two values $\boldsymbol{v}_1 \in \mathcal{M}_1$ and $\boldsymbol{v}_2 \in \mathcal{M}_2$ are equal (with respect to φ) if $\boldsymbol{v}_2 = \varphi(\boldsymbol{v}_1)$.
Also let $B_1, B_2 \in \mathbb{R}^+$ and Unv_1, Unv_2 be the unveil functions corresponding to the respective commitment functions with bound B_1 and B_2 respectively, as described in Figure 1. Furthermore, let $\mathsf{Unv}_{1,2B_1}$, and $\mathsf{Unv}_{2,2B_2}$ denote the unveil functions corresponding to the respective commitment functions with bound $2B_1$ and $2B_2$ respectively.
Finally, let $\mathsf{Com}_{1,a}$ and $\mathsf{Com}_{2,a}$ be two auxiliary computationally binding and statistically hiding commitment schemes with message space $\mathbb{Z}_{q_1}^{m_1}$ and $\mathbb{Z}_{q_2}^{m_2}$ respectively, and $\mathsf{Unv}_{1,a}$ and $\mathsf{Unv}_{2,a}$ be the corresponding unveil functions.
Figure 5 shows the proof of message equality protocol for LPCom.

The public input is $(\boldsymbol{c}_1, \boldsymbol{c}_2) \in \mathbb{Z}_{q_1}^{m_1} \times \mathbb{Z}_{q_2}^{m_2}$ and the private input is $(\boldsymbol{v}_1, \boldsymbol{r}_1, \boldsymbol{r}_2) \in \mathcal{M}_1 \times \mathbb{Z}_{q_1}^{k_1} \times \mathbb{Z}_{q_2}^{k_2}$.

1. P secretly samples $\boldsymbol{v}_1' \xleftarrow{\$} \mathcal{M}_1$ and sets $\boldsymbol{v}_2' := \varphi(\boldsymbol{v}_1')$. Then for all $i \in \{1,2\}$ it does the following. P computes $\boldsymbol{e}_i = \boldsymbol{c}_i - \boldsymbol{A}_{i,1}\boldsymbol{v}_i - \boldsymbol{A}_{i,2}\boldsymbol{r}_i$. Then P secretly samples $\boldsymbol{r}_i' \xleftarrow{\$} \mathbb{Z}_{q_i}^{k_i}$, $\boldsymbol{e}_i' \xleftarrow{\$} D_{\sigma_i'}^{m_i}$ and computes the commitment $\boldsymbol{c}_i' = \mathsf{com}(\boldsymbol{v}_i', \boldsymbol{r}_i', \boldsymbol{e}_i')$. P then computes two auxiliary commitments $\boldsymbol{c}_{i,a}'$ and corresponding opening randomness values $\boldsymbol{r}_{i,a}'$ using $\mathsf{Com}_{i,a}$ with input $\boldsymbol{c}_i'$. Then P sends $(\boldsymbol{c}_{1,a}', \boldsymbol{c}_{2,a}')$ to V.
2. V sends a random bit $b \in \{0,1\}$ to P.
3. P computes $\boldsymbol{v}_{i,b}' = \boldsymbol{v}_i' + b\boldsymbol{v}_i$ and $\boldsymbol{r}_{i,b}' = \boldsymbol{r}_i' + b\boldsymbol{r}_i$ for all $i \in \{1,2\}$, where $\boldsymbol{v}_2 = \varphi(\boldsymbol{v}_1)$.
   if $b = 0$:
   P sends $(\boldsymbol{c}_1', \boldsymbol{r}_{1,a}', \boldsymbol{v}_{1,b}', \boldsymbol{r}_{1,b}', \boldsymbol{c}_2', \boldsymbol{r}_{2,a}', \boldsymbol{v}_{2,b}', \boldsymbol{r}_{2,b}')$ to V.
   if $b = 1$:
   With probability
   $$p = \min\left(\frac{D_{\sigma_1'}^{m_1}(\boldsymbol{e}_1')}{M_1 D_{\boldsymbol{e}_1, \sigma_1'}^{m}(\boldsymbol{e}_1')}, 1\right) \cdot \min\left(\frac{D_{\sigma_2'}^{m_2}(\boldsymbol{e}_2')}{M_2 D_{\boldsymbol{e}_2, \sigma_2'}^{m}(\boldsymbol{e}_2')}, 1\right),$$
   P sends $(\boldsymbol{c}_1', \boldsymbol{r}_{1,a}', \boldsymbol{v}_{1,b}', \boldsymbol{r}_{1,b}', \boldsymbol{c}_2', \boldsymbol{r}_{2,a}', \boldsymbol{v}_{2,b}', \boldsymbol{r}_{2,b}')$ to V, and $\perp$ otherwise.
4. V accepts iff P did send $\perp$ and $\boldsymbol{v}_{2,b}' = \varphi(\boldsymbol{v}_{1,b}')$ and $\mathsf{Unv}_{i,a}(\boldsymbol{c}_{i,a}', \boldsymbol{c}_i', \boldsymbol{r}_{i,a}') = \boldsymbol{c}_i'$ and $\boldsymbol{v}_{i,b}' \in \mathcal{M}_i$ and $\mathsf{Unv}_i(\boldsymbol{c}_i' + b\boldsymbol{c}_i, \boldsymbol{v}_{i,b}', \boldsymbol{r}_{i,b}') = \boldsymbol{v}_{i,b}'$ for all $i \in \{1,2\}$.

Fig. 5. Proof of Message Equality

Theorem 8 *With the indexed parameters as given in Figure 4 and sufficiently large security parameters k_1 and k_2 the protocol defined in Figure 5 is a Σ^*-protocol with completeness error overwhelmingly close to $\frac{1}{2} - \frac{1}{2M_1M_2}$ for the relations*

$$\begin{aligned}\mathcal{R} = \{((\boldsymbol{c}_1, \boldsymbol{c}_2), (\boldsymbol{v}_1, \boldsymbol{r}_1, \boldsymbol{r}_2)) \in (\mathbb{Z}_{q_1}^{m_1} \times \mathbb{Z}_{q_2}^{m_2}) \times (\mathcal{M} \times \mathbb{Z}_{q_1}^{k_1} \times \mathbb{Z}_{q_2}^{k_2}) \mid \\ \boldsymbol{v}_1 = \mathsf{Unv}_1(\boldsymbol{c}_1, \boldsymbol{v}_1, \boldsymbol{r}_1) \wedge \varphi(\boldsymbol{v}_1) = \mathsf{Unv}_2(\boldsymbol{c}_2, \varphi(\boldsymbol{v}_1), \boldsymbol{r}_2)\} \text{ and} \\ \mathcal{R}' = \{((\boldsymbol{c}_1, \boldsymbol{c}_2), (\boldsymbol{v}_1, \boldsymbol{r}_1, \boldsymbol{r}_2)) \in (\mathbb{Z}_{q_1}^{m_1} \times \mathbb{Z}_{q_2}^{m_2}) \times (\mathcal{M} \times \mathbb{Z}_{q_1}^{k_1} \times \mathbb{Z}_{q_2}^{k_2}) \mid \\ \boldsymbol{v}_1 = \mathsf{Unv}_{1,2B_1}(\boldsymbol{c}_1, \boldsymbol{v}_1, \boldsymbol{r}_1) \wedge \varphi(\boldsymbol{v}_1) = \mathsf{Unv}_{2,2B_1}(\boldsymbol{c}_2, \varphi(\boldsymbol{v}_1), \boldsymbol{r}_2)\}.\end{aligned}$$

Proof: Similar to the proof of Theorem 7, since this is essentially the protocol defined in Figure 3 run twice in parallel. ■

As in the proof of knowledge, it is necessary to repeat the message equality proof often enough to amplify the success probability of an honest and a dishonest prover. Similar to the analysis presented in Section 4.3, we propose to accept the prover if it answers at least $N/2 \cdot (1 + 1/(2M_1M_2))$ challenges correctly, and estimate the necessary number of rounds to be $N = 4k(M_1M_2)^2$.

5.2 Prolonging the Bindingness

In the prolonging algorithm presented in Figure 6 we show how to prolong the bindingess of a commitment $\boldsymbol{c}_1$ to a message $\boldsymbol{v}$. Possible instantiations of LPCom that can be used for prolonging are ones according to Figure 4. However, since other parameter choices are possible we give the more general result here.

The first step to prolong the bindingness of a commitment $\boldsymbol{c}_1$ is to generate a new commitment $\boldsymbol{c}_2$ to the same (or an equivalent) value. Two values are considered equivalent if a signalized isomorphism between the two message spaces maps one value to the other. A possible way to achieve isomorphic message spaces is to use message spaces isomorphic to $\mathbb{Z}_2^n$, i.e. $\mathcal{M} = \{0, q/2\}^2$ (see Remark 4 for more details).

In Section 3.3 we showed that violating the bindingness of LPCom instantiated with parameters according to Figure 2 gets increasingly harder with increasing security parameter k. It is easy to see that the same is true if LPCom is instantiated according to Figure 4. If both schemes are instantiated according to Figure 4, violating the binding property of the new commitment (instantiated with security parameter k_2) is in fact harder than violating the bindingness of the old commitment (instantiated with security parameter k_1) as long as $k_2 > k_1$.

Let Com_1 (and the corresponding unveil function) be a correct, statistically hiding, and computationally binding instantiation of LPCom such that the following holds: When using Com_1, the protocol for proving message equality defined in Figure 5 is a Σ^*-protocol. The procedure to prolong the bindingness to a commitment $\boldsymbol{c}_1 = \mathsf{Com}(\boldsymbol{v})$ works as described in Figure 6.

1. Choose a new security parameter k_2 for the desired security level.
2. Create a new instantiation Com_2 of LPCom with security parameters κ, k_2 and parameters (e.g. according to Figure 4) such that the following conditions are satisfied:
   - Com_2 is correct, statistically hiding, and computationally binding.
   - The message space $\mathcal{M}_2$ of Com_2 is isomorphic to the message space $\mathcal{M}_1$ of Com_1.
   - When using Com_2, the proof of knowledge protocol defined in Figure 5 is a Σ^*-protocol.
3. Specify an isomorphism $\varphi : \mathcal{M}_1 \to \mathcal{M}_2$. (See Remark 4.)
4. The committer creates a new commitment $\boldsymbol{c}_2$ to the value $\varphi(\boldsymbol{v})$ using Com_2.
5. Using the protocol defined in Figure 5, the committer shows in statistical zero-knowledge fashion that it can open $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ to the same value $\boldsymbol{v}$ (or $\varphi(\boldsymbol{v})$) with the corresponding unveil functions.

Fig. 6. Protocol for Prolonging the Bindingness

Remark 4 *Note that the isomorphism φ needs to be publicly stored with the commitment in order to extract the old value $\boldsymbol{v}$ from the new value $\varphi(\boldsymbol{v})$. Alternatively, one can choose the message spaces such that there is a canonical isomorphism between them and specify "canonical" in the prolonging protocol. We now provide one possible canonical choice. Always let the message spaces $\mathcal{M} \subseteq \mathbb{Z}_q^n$ be of the form $\mathcal{M} = G^n$, where $G = \{0, \frac{q}{2}\}$ is the (uniquely determined) subgroup of $\mathbb{Z}_q$ of order two. Note that q is required to be even for this to be possible. If now $\mathcal{M}_1 = G_1^n$ and $\mathcal{M}_2 = G_2^n$ are of this form, then there is a unique isomorphism $\varphi' : G_1 \to G_2$ mapping the non-zero element of G_1 to the non-zero element of G_2. Applying this isomorphism in every coordinate, φ' naturally extends to an isomorphism $\varphi : G_1^n \to G_2^n$.*

6 Switching from Pedersen Commitment to LPCom

Up until now in practice usually Pedersen commitments are used when an unconditionally hiding commitment scheme is required. This is due to the fact that this commitment scheme is very efficient. In [10], the authors show that the bindingness of Pedersen commitments can also be prolonged. However, this is only secure as long as the discrete logarithm problem remains unbroken, in particular not in the presence of a quantum adversary. Thus, in this section we want to discuss how to proceed if a Pedersen commitment must be replaced by a prolongable post-quantum commitment, i.e. by the construction presented in this work.

We discuss the following approach to switch from Pedersen commitments to our lattice construction. One creates a new commitment using our lattice construction and shows in statistical zero-knowledge fashion that one can open the lattice and the Pedersen commitment to the same value. In order for this to work, one needs a protocol for proving message equality of two commitments, where one is created using our construction and the other using the Pedersen construction. Since our protocol and the protocol for proving message equality for Pedersen commitments follow essentially the same idea (see [20]), this should be a straightforward generalization of our protocol proposed in Figure 5. Remember that for the protocol to work it is crucial that the underlying massage spaces are isomorphic. In the following, we argue that it is possible to use isomorphic message spaces in both constructions.

In Pedersen commitments, the message space $\mathcal{M}_P$ is essentially a cyclic group of order p, where p is a sufficiently large prime. In our lattice construction the message space $\mathcal{M}_L$ is a subgroup of $\mathbb{Z}_q^n$ for some integers q and n. In order for $\mathcal{M}_P$ and $\mathcal{M}_L$ to be isomorphic, we must have that $|\mathcal{M}_L| = |\mathcal{M}_P| = p$. Since of course the order of the subgroup $|\mathcal{M}_L|$ divides the group order $|\mathbb{Z}_q^m| = q^m$, we obtain $p \mid q^m$, hence $p \mid q$, since p is prime. In particular we have $p \leq q$. Since typical values for p are much larger than typical lattice moduli q, the most efficient choice is setting $q := p$ and $\mathcal{M}_L = \mathbb{Z}_q$.

Besides being able to "convert" a Pedersen commitment to a lattice-based commitment, this method also allows to directly generate two commitments

to the same value v: one generated using the Pedersen commitment scheme and the second one using LPCom. The advantage of this approach is that the bindingness of the commitment relies on two hardness problems, namely the discrete logarithm problem and α-SVP. Notice that since the hidingness of both commitment schemes is unconditionally, there is no disadvantage regarding the hiding property. On the downside this instantiation is very inefficient, at least with respect to the lattice commitment. This is because typical lattice moduli q have size about 13 bit, while usually the group order p of the message space in a Pedersen scheme is of size about 190 bits (see [10]). However, this is a first step in the direction of double hardness for commitment schemes and we leave the task to find more efficient constructions for future work. One promising direction is to combine schemes using ideal lattices [18] with Pedersen commitments as shown for encryption schemes by [3].

7 Conclusion and Future Work

In this paper, we proposed and assessed the security of LPCom, a new lattice-based, unconditionally hiding commitment scheme with prolongable computational bindingness. In addition, we provided two asymptotic parameter sets: one tuned for efficiency and a second one that supports proofs of knowledge and message equality. Since LPCom is based on α-SVP, it is reasonable to view it as a good building block for long-term secure cryptographic services, e.g. digital document archiving, in the event quantum computing becomes practical. Furthermore, through a simple security prolonging procedure we are able to convert a Pedersen commitment into a lattice-based commitment using LPCom.

For future work we plan to improve the efficiency of our construction. One promising direction is to use ideal lattices in our approach, which are known to increase efficiency [18, 11]. Note that our construction already performs well since the commitment and opening value sizes are quasi-linear in the security parameter. However, in [3] Benhamouda et al. show how to build an efficient "hybrid" scheme containing two primitives - Pedersen commitments and an ideal-lattice-based encryption scheme - such that it can be shown that both are committing to the same message. Although LPCom already provides this feature, our approach to build this "hybrid" scheme is still quite inefficient. Furthermore, we would like to more closely analyse the homomorphic properties of LPCom. Even though the commitment function is additively homomorphic care should be taken when working on the commitments. The more commitments are added, the bigger the error term grows within the result, and the more likely the unveil algorithm does not accept the input as valid, meaning the resulting commitment cannot be opened anymore. Therefore, the restriction of this function must be analysed. Another observation is that in a sum of two commitments the resulting error term leaks information about the error terms of each summand. Thus, it must be determined how much information is revealed and how to deal with this depending on the respective application. Nevertheless, our construction is a very

promising building block for long-lived systems and we plan to work on these matters in the future.

Acknowledgments

This work has been co-funded by the DFG as part of project Future Public-Key Encryption and Signature Schemes and Long-Term Secure Archiving within the CRC 1119 CROSSING. In addition, this project has received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreement No 644962.

References

1. Ambainis, A., Rosmanis, A., Unruh, D.: Quantum attacks on classical proof systems: The hardness of quantum rewinding. In: FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014. pp. 474–483 (2014)
2. Bellare, M., Goldreich, O.: On defining proofs of knowledge. In: Advances in Cryptology - CRYPTO '92, Santa Barbara, California, USA, August 16-20, 1992, Proceedings. pp. 390–420 (1992)
3. Benhamouda, F., Camenisch, J., Krenn, S., Lyubashevsky, V., Neven, G.: Better zero-knowledge proofs for lattice encryption and their application to group signatures. In: Advances in Cryptology - ASIACRYPT 2014, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. pp. 551–572 (2014)
4. Benhamouda, F., Krenn, S., Lyubashevsky, V., Pietrzak, K.: Efficient zero-knowledge proofs for commitments from learning with errors over rings. IACR Cryptology ePrint Archive 2014, 889 (2014), `http://eprint.iacr.org/2014/889`
5. Canetti, R., Halevi, S., Katz, J., Lindell, Y., MacKenzie, P.: Universally composable password-based key exchange. In: Cramer, R. (ed.) Advances in Cryptology, EUROCRYPT 2005, pp. 404–421. Springer Berlin Heidelberg (2005)
6. Chaum, D., Pedersen, T.P.: Wallet databases with observers. In: Advances in Cryptology - CRYPTO '92, Santa Barbara, California, USA, August 16-20, 1992. pp. 89–105 (1992)
7. Cramer, R.: Modular Design of Secure Yet Practical Cryptographic Protocols (1997), `https://books.google.de/books?id=kMZcAAAACAAJ`
8. Damgard, I.: Commitment schemes and zero-knowledge protocols. In: Damgard, I. (ed.) Lectures on Data Security, Lecture Notes in Computer Science, vol. 1561, pp. 63–86. Springer Berlin Heidelberg (1999)
9. Damgård, I.: On $\sigma-$Protocols (2010)
10. Demirel, D., Lancrenon, J.: How to securely prolong the computational bindingness of pedersen commitments. Cryptology ePrint Archive, Report 2015/584 (2015), `http://eprint.iacr.org/`
11. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: Advances in Cryptology - CRYPTO, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. pp. 40–56 (2013)
12. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Advances in Cryptology - EUROCRYPT 2008, Istanbul, Turkey, April 13-17, 2008. Proceedings. pp. 31–51 (2008)

13. Gentry, C., Peikert, C., Vaikuntanathan, V.: 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. pp. 197–206 (2008)
14. Haber, S.: Content Integrity Service for Long-Term Digital Archives. In: Archiving 2006. pp. 159–164. IS&T, Ottawa, Canada (2006)
15. Lindner, R., Peikert, C.: Better key sizes (and attacks) for lwe-based encryption. In: Topics in Cryptology - CT-RSA 2011, San Francisco, CA, USA, February 14-18, 2011. pp. 319–339 (2011)
16. Lyubashevsky, V.: Lattice-based identification schemes secure under active attacks. In: Public Key Cryptography - PKC 2008, Barcelona, Spain, March 9-12, 2008. Proceedings. pp. 162–179 (2008)
17. Lyubashevsky, V.: Lattice signatures without trapdoors. In: Advances in Cryptology - EUROCRYPT 2012 Cambridge, UK, April 15-19, 2012. Proceedings. pp. 738–755 (2012)
18. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. J. ACM 60(6), 43 (2013), http://doi.acm.org/10.1145/2535925
19. Micciancio, D., Regev, O.: Worst-case to average-case reductions based on gaussian measures. SIAM J. Comput. 37(1), 267–302 (2007), http://dx.doi.org/10.1137/S0097539705447360
20. Moran, T., Naor, M.: Split-ballot voting: Everlasting privacy with distributed trust. ACM Trans. Inf. Syst. Secur. 13(2) (2010)
21. Pass, R.: Limits of provable security from standard assumptions. In: Proceedings of STOC 2011, San Jose, CA, USA, 6-8 June 2011. pp. 109–118 (2011)
22. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: CRYPTO. pp. 129–140 (1991)
23. Schnorr, C.: Efficient signature generation by smart cards. J. Cryptology 4(3), 161–174 (1991), http://dx.doi.org/10.1007/BF00196725