Sharp: Short Relaxed Range Proofs

Geoffroy Couteau[1], Dahmun Goudarzi[2], Michael Klooß[3], and Michael Reichle[4]

[1]CNRS, IRIF, Université de Paris, France, couteau@irif.fr
[2]Unaffiliated, dahmun.goudarzi@gmail.com
[3]Karlsruhe Institute for Technology, KASTEL, michael.klooss@kit.edu
[4]DIENS, École normale supérieure, PSL University, CNRS, INRIA, Paris, France, michael.reichle@ens.fr

September 6, 2022

We provide optimized range proofs, called Sharp, in discrete logarithm and hidden order groups, based on square decomposition. In the former setting, we build on the paradigm of Couteau et al. (Eurocrypt '21) and optimize their range proof (from now on, CKLR) in several ways: (1) We introduce batching via vector commitments and an adapted Σ-protocol. (2) We introduce a new group switching strategy to reduce communication. (3) As repetitions are necessary to instantiate CKLR in standard groups, we provide a novel batch shortness test that allows for cheaper repetitions. The analysis of our test is nontrivial and forms a core technical contribution of our work. For example, for $\lambda = 128$ bit security and $B = 64$ bit ranges for $N = 1$ (resp. $N = 8$) proof(s), we reduce the proof size by 34% (resp. 75%) in arbitrary groups, and by 66% (resp. 88%) in groups of order 256-bit, compared to CKLR.

As Sharp and CKLR proofs satisfy a "relaxed" notion of security, we show how to enhance their security with *one* additional hidden order group element. In RSA groups, this reduces the size of state of the art range proofs (Couteau et al., Eurocrypt '17) by 77% ($\lambda = 128, B = 64, N = 1$).

Finally, we implement our most optimized range proof. Compared to the state of the art Bulletproofs (Bünz et al., S&P 2018), our benchmarks show a very significant runtime improvement. Eventually, we sketch some applications of our new range proofs.

1. Introduction

Zero-Knowledge Proofs and Range Proofs. Zero-knowledge proofs, introduced in the seminal work of Goldwasser, Micali, and Rackoff [GMR89], allow a prover to convince a verifier of the truth of a statement while concealing all other information. This makes them an important tool in theory and practice. Efficient constructions are now known for a variety of NP-languages, and are routinely used in real-world applications. An example of particular interest is range proofs, which are zero-knowledge proofs for demonstrating that a secret value (committed or encrypted) belongs to a public range. Range proofs are a core component in numerous applications, such as

anonymous credentials [Cha90], e-voting [Gro05], or e-cash [CHL05], and have been introduced recently in some popular anonymous cryptocurrencies (see [Zca; Mon; Bün+20]).

Range Proofs. Many range proofs which have been constructed in the past can be categorized in two main paradigms:

(1) Range proofs based on n-ary decomposition [CCs08; Gro11], where one proves a statement of the form $x \in [0, n^\ell)$ by committing to an n-ary decomposition $(x_0, \ldots, x_{\ell-1})$ of x, and proving that $x = \sum_i x_i \cdot n^i$ and each x_i belongs to $[0, n)$ (which can be done efficiently when n is small). The state of the art method in this paradigm is Bulletproofs [Bün+18], which features very small proof size $O(\lambda \cdot \log \ell)$ for a security parameter λ (using binary decomposition), and also enjoys a transparent setup: the only trusted parameter it requires is an unstructured common random string, which can be easily generated by standard "nothing up my sleeve" methods (in contrast, protocols requiring a structured common string need to trust the parameter generator, which is undesirable). Due to its great concrete efficiency and its transparent setup, Bulletproofs have become the most commonly used solution in real-world applications.

(2) Range proofs based on square decomposition [Bou00; Lip03; Gro05; CPP17], where one proves a statement of the form $x \geq 0$ by using special *integer commitment schemes* [FO97; DF02] to commit to x over $\mathbb{Z}$, and by proving the existence of four squares $x_1, \ldots, x_4$ such that $x = \sum_i x_i^2$ (such a decomposition always exist by a theorem of Lagrange, and ensures non-negativity). This generalizes to arbitrary intervals $[a, b]$ by proving non-negativity of $(x-a)(b-x)$. While avoiding n-ary decomposition is attractive, instantiating integer commitments required until recently the use of hidden order groups (such as RSA groups), whose elements are too large to be competitive with Bulletproofs for any reasonable interval size, and which require a trusted setup (to set up the RSA modulus).

The CKLR Range Proof. In a recent work [Cou+21a], Couteau *et al.* revived the square decomposition paradigm, by constructing *bounded* integer commitment schemes, which can be instantiated over cryptographic groups with hard DLOG problem. They instantiate (a variant of) the range proof of [CPP17] with this new commitment scheme, significantly reducing their size and removing the need for a structured common reference string. The CKLR scheme was shown to compare favorably with Bulletproofs: for a careful choice of parameters and underlying group, the proofs are about 15% shorter than Bulletproofs, and require an order of magnitude less group operations. Therefore, on paper, CKLR seems to offer a competitive alternative to Bulletproofs.

CKLR versus Bulletproofs. However, this cost estimation ignores several important practical aspects, and the distinction turns out to be far from clear cut in real-world instantiations. The main limitation of CKLR is that it requires exotic group sizes – typically, elliptic curves with elements of size 352 or 416 bits to achieve 128 bits of security for 32- or 64-bit ranges. While in theory, we can use curves with a wide variety of sizes, and many standard options exist, the vast majority of cryptographic applications build upon 256-bit elliptic curves, and highly optimized implementations of some of these curves are available (for example in libsecp256k1 [Wui18] or ristretto255 [Val+19]). These libraries typically offer runtimes 10 to 20 times faster than the NIST standardized implementations of other standard curves. Hence, the use of large curves in CKLR actually negates the efficiency gains of their smaller number of group operations compared to Bulletproofs. Furthermore, several applications constrain the choice of curve; for example, the Ethereum cryptocurrency only allows the curve secp256k1.

This is not the only limitation of the CKLR range proof, compared to Bulletproofs. The latter is especially attractive when performing several range proofs at once, because it allows for very efficient batching of multiple proofs; no such batching is known for CKLR. This stems from the

fact that the CKLR range proof revolves around an "extraction lemma" which was formulated and proven in the setting of a single proof, and operates on top of single-value commitments (while Bulletproofs operate on generalized Pedersen commitments, which can commit compactly to vectors of values).

Eventually, CKLR is also more restricted in its range of applications compared to Bulletproofs. This is because Bulletproofs operate with standard Pedersen commitments, while CKLR is designed on top of a new (Pedersen-based) construction of bounded integer commitments. Compared to Pedersen commitments, these new commitments have (1) only limited homomorphic properties, and (2) a relaxed notion of opening, where a malicious opener is given more freedom in what is regarded as a valid opening (this is similar in spirit to the property of standard integer commitment schemes, such as the Damgård-Fujisaki commitment [DF02]). This means that in some applications, for example when a value opened by a malicious party must be reused afterwards by an honest prover (this is the case, e.g. in some cryptocurrency applications), CKLR cannot be used as a drop-in replacement: the use of CKLR is only appropriate when the new commitment scheme can be used in the application without harming security or correctness.

Summing up, the CKLR paradigm is a promising new approach for constructing range proofs with strong performance. However, it does not currently compare favorably to Bulletproofs in practical applications, mostly due to its use of larger curves which lack competitive implementations, but also due to its lack of batching features. Furthermore, it operates on a new commitment scheme, which makes it not a priori clear what are the standard applications of range proofs where it can be safely used.

1.1. Our Contributions

In this work, we thoroughly revisit the CKLR paradigm. We introduce a new family of range proof schemes, which we call Sharp (for short relaxed range proofs). The name Sharp stems from a change of perspective with respect to CKLR: in CKLR, a proof is interpreted as a full-fledged range proof for values committed with a new *bounded integer commitment* which they introduce. The latter is essentially a Pedersen commitment where openings are allowed to be *rationals*, which are rounded to the nearest integer in the opening phase. We observe that one can equivalently "push the relaxation from the commitment to the range proof" and see CKLR as a *relaxed* range proof operating over standard Pedersen commitments, where *relaxed* means that the prover is only bound to a *rational* inside the target range, instead of an integer.[1] While this change of perspective does not in itself change the construction nor its security properties, it allows for a more modular treatment of the construction, and simplifies the analysis of how CKLR (or Sharp) integrates within standard application of range proofs.

Our new constructions build upon numerous optimizations, which are a combination of known techniques and entirely new approaches. The security analysis of our scheme is subtle and technically involved; it forms the core technical contribution of our work. Sharp proofs improve upon CKLR on all possible fronts: they are much shorter, more efficient, allow for a considerably more flexible choice of the underlying group (and can in particular be efficiently instantiated over 256-bit curves), and can be batched efficiently. In addition, we also demonstrate how to overcome the relaxation of soundness, obtaining schemes that operate directly with standard Pedersen commitments and effectively bind the prover to an *integer* in the range (instead of a rational) at the cost of slightly larger proofs (but still with very competitive performance).

To complement the above results, we elaborate on how Sharp can be used to improve the efficiency of some flagship applications of range proofs, such as anonymous credentials and anonymous transactions, clarifying which applications can work with bounded integer commitment

[1] This is a purely conceptual change of view with respect to CKLR, where the rational opening is afterwards interpreted as an encoding of the closest integer via rounding.

schemes, and which require using a scheme with stronger features. We validate our efficiency claims with implementations and benchmarks of our main schemes. While our implementation is an unoptimized proof-of-concept implementation, our benchmarks show that it offers a ten-fold runtime improvement over a heavily optimized implementation of Bulletproofs; we expect that the efficiency gap would widen further with a more optimized implementation of Sharp. Below, we elaborate on our contributions.

1.1.1. Improved Range Proof Constructions.

Our new family of range proofs, Sharp, can be instantiated in a variety of settings, leading to tradeoffs between efficiency and the underlying soundness notion. We build upon the paradigm introduced in [Cou+21a] and obtain range proofs with improved efficiency and flexibility. In applications where low communication matters the most, our scheme $\mathsf{Sharp}_{\mathsf{GS}}$ provides the most competitive performance, but uses curves of sizes other than the standard 256-bit setting. For runtime-critical applications, or when the application restricts the available curve, we describe $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$, a scheme fully optimized to work over 256-bit groups.

At the heart of our flexibility and efficiency improvements is a modular treatment of the structure of a range proof. We split the range proof into two conceptual parts: the proof of short opening (PoSO) and the proof of decomposition (PoDec). The PoSO guarantees that extracted openings are short and the PoDec ensures that the square decomposition holds over $\mathbb{Z}_p$, where p is the order of the DLOG group. Combining both parts ensures that the committed value is a *rational* inside the given range, as the shortness allows us to argue over the integers. This decoupling allows us to develop tailored optimizations for each part, but also clarifies the exact soundness guarantees which the proof provides. We stress that one can still equivalently see Sharp as a standard range proof operating over a relaxed integer commitment scheme, using the rounding technique of CKLR: our change of perspective improves the conceptual simplicity of analyzing the use of Sharp within standard applications, but the exact guarantees remain identical to CKLR.

Optimizing the decomposition proof. We optimize the PoDec via a polynomial-based technique, similar to the lattice version of [Cou+21a] (with some tweaks that improve efficiency). Besides improving efficiency of the PoDec, this adaption enables two additional improvements: (1) The new protocol is suited for vector commitments, such as Pedersen multi-commitments (MPed). This enables more efficient batch range proofs, in the sense of performing range proofs for all N values in the vector commitment at once. (2) We introduce a group switching strategy that enables the use of different groups for the PoSO and PoDec. To our knowledge, this is the first time group switching is (efficiently) used without leveraging hidden order groups. This optimization further reduces proof length (and computation), while allowing more flexibility to instantiate the underlying groups. These changes lead to an optimized range proof: $\mathsf{Sharp}_{\mathsf{GS}}$.

Optimizing the short opening proof. We further present $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$, a range proof with optimized PoSO (in combination with the changes described above). The analysis of this scheme is delicate and uses several new ideas. It constitutes the main technical contribution of this work. As range and challenge space (hence soundness) introduce lower bounds on group size, repetitions are required to achieve high security levels when the group is fixed. In CKLR, such repetitions were very expensive, as much of the proof had to be repeated. To reduce their cost, we introduce a (fractional) shortness test that allows the prover to show that numerator and denominator of multiple fractions are *short* by sending a single *short* integer, per repetition. Integrating this shortness test in the range proof, a "repetition" requires only two scalars, independent of the batch size. Thus, the bulk of communication and computation of the range proof is the

optimized PoDec (*without* any repetition).

We note that these optimizations also lead to significant improvements in a batch setting, where multiple range proofs must be executed at once. For example, executing $N = 8$ range proofs with 128 bits of security and 64-bit inputs communicates only 2.9 times more than executing a single range proof. We also observe that a similar batch technique is used in the context of lattice-based range proofs, in the setting where all challenges are bits. However, the possibility of using general short challenges instead of bits is precisely what allows our schemes to remain very compact, and is also what makes the analysis of our shortness test so delicate (we elaborate on this aspect in the technical overview).

Binding to integers instead of rationals. The bounded integer commitment scheme of [Cou+21a] is essentially a Pedersen commitment where malicious openers are allowed to reveal a rational instead of an integer (that is later rounded to encode an integer inside the range). Consequently $\mathsf{Sharp}_{\mathsf{GS}}$, like CKLR, provides only a relaxed notion of soundness, in that it only binds the prover to a rational in the target range. We develop several new approaches to overcome this limitation, obtaining proofs that operate with standard Pedersen commitments (where openings are required to be integers). In the interactive setting, where soundness is statistical (and a 2^{-40} statistical soundness error is a common choice), we show how our batch shortness test allows us to use challenges in $\{0, 1\}$ with much more reasonable communication overhead compared to previous approaches, which gives a competitive three-round range proof with transparent setup and full-fledged soundness. In the non-interactive setting (where soundness is computational and 128 repetitions would be too expensive), we show how to combine our schemes with a minimal use of hidden order groups, obtaining two variants: $\mathsf{Sharp}_{\mathsf{CL}}$ (using class groups to instantiate the hidden order group) and $\mathsf{Sharp}_{\mathsf{RSA}}$ (using RSA groups). These variants retain a strong efficiency, as only a *single element* of the hidden order group must be added to the proof. They achieve stronger soundness notions, namely: (1) $\mathsf{Sharp}_{\mathsf{RSA}}$ achieves standard soundness (allowing our scheme to be used as a drop-in replacement in essentially any application of range proofs, but at the cost of loosing the transparent setup), and (2) $\mathsf{Sharp}_{\mathsf{CL}}$ achieves a slightly weaker soundness where the prover is bound to a *dyadic rational*, which suffices to overcome some attacks that arise from the use of a range proof with relaxed soundness in some applications, while retaining the transparent setup.

We note that many range proofs in RSA groups have been described in the past [Bou00; Lip03; Gro05; CPP17]. Our RSA-based variant achieves considerable efficiency improvements compared to all these previous works (both communication and computation-wise), while achieving the same soundness guarantees.

Concrete efficiency estimations. We compare the communication efficiency of $\mathsf{Sharp}_{\mathsf{GS}}$, $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$, and $\mathsf{Sharp}_{\mathsf{RSA}}$ to the state-of-the-art in table 1. For performing a single range proof, $\mathsf{Sharp}_{\mathsf{GS}}$ proofs are almost 50% shorter than Bulletproofs, and about 34% shorter than the CKLR range proofs. For our computation-optimized range proofs $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$, these numbers are about 42% and 29% respectively. When performing a large number of range proofs, Bulletproofs become better communication-wise, because of their logarithmic cost in the batch size; nevertheless, even for a batch of $N = 8$ range proofs, our range proofs are only between 1.1 and 1.3 times larger than Bulletproofs (in concrete applications, we believe that this should be largely compensated by our strong computational improvements). Our variant in RSA groups, which achieves standard soundness, improves by a large margin compared to the previous best-known RSA-based range proof of [CPP17]: a factor 3 improvement for a single range proof, and up to a factor 14 improvement for $N = 8$ simultaneous range proofs.

We implemented our computation-optimized range proof $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$, using the 256-bit elliptic curve from the libsecp256k1 library [Wui18]. We stress that this is an unoptimized implementation;

yet, compared to the optimized reference implementation of Bulletproofs using the same library, and running the two protocols on the same machine, we observe very significant runtime improvements. The runtime of our prover is 11 to 17 times faster than Bulletproofs' (for 32-bit and 64-bit ranges), while our verifier is two to four times faster; see table 2. For a larger batch size of $N = 8$, our verifier runtime remains two to four times faster than Bulletproofs, while the gap with our prover runtimes increases slightly, ranging from 11 to 21 times faster (all while maintaining a proof size only 1.1 to 1.3 larger than that of Bulletproofs for $N = 8$). We expect these gaps to further increase with a more optimized implementation.

Table 1: Theoretical proof size in Bytes for showing that some $x \in [0, B]$ of CKLR proofs [Cou+21a], Bulletproofs [Bün+18], RSA-based range proofs [CPP17] and Sharp proofs ($\mathsf{Sharp}_{\mathsf{GS}}$, $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ and $\mathsf{Sharp}_{\mathsf{RSA}}$) given the security parameter λ. The groups $\mathbb{G}_{\mathsf{com}}$ and $\mathbb{G}_{\mathsf{3sq}}$ used for Sharp proofs have order p and q respectively. π denotes proof size in Bytes, N denotes the number of proofs in the batch, and $\log p, \log q$ is the bit-size of p and q.

		CKLR		BPs	RSA	$\mathsf{Sharp}_{\mathsf{GS}}$			$\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$			$\mathsf{Sharp}_{\mathsf{RSA}}$
$(\lambda, \log B)$	N	$\log p$	π	π	π	$\log p$	$\log q$	π	$\log p$	$\log q$	π	π
128, 64	1	416	545	672	2424	333	411	360	256	256	389	793
	8	416	4360	864	19056	333	411	1070	256	256	1119	1503
	16	416	8720	928	38064	333	411	1882	256	256	1928	2315
128, 32	1	352	501	608	2404	301	347	318	256	256	335	751
	8	352	4008	800	18896	301	347	916	256	256	932	1349
	16	352	8016	864	37744	301	347	1600	256	256	1612	2033

Table 2: Benchmark of our optimized range proofs compared to Bulletproofs, using the reference Bulletproofs implementation in C of [Bün+18], using batch sizes $N = 1$ and $N = 8$. Both implementations use the library libsecp256k1 [Wui18], and were run on a MacBook Pro with a 2.3 GHz Intel core i7 processor. All timings are in milliseconds.

		Bulletproofs		$\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$	
$(\lambda, \log B)$	N	Prover's work	Verifier's work	Prover's work	Verifier's work
128, 64	1	20.6	2.55	1.17	0.75
	8	157	12.1	7.47	3.88
128, 32	1	10.5	1.46	0.97	0.74
	8	80.0	6.93	6.74	3.39

1.1.2. Security and Applications.

We analyze the guarantees of range proofs with relaxed soundness (such as CKLR and Sharp) in standard range proof applications. For this, we show which manipulations of the committed values can be allowed depending on the setting. Specifically, we discuss the arithmetical behaviour of the manipulated rationals, the impact of the chosen decomposition on soundness and show that Sharp proofs provide standard soundness when the committed values are short. Then, we use these insights to sketch how Sharp can be applied to two important applications of range proofs: anonymous credentials (AC) and anonymous transactions (AT). While relaxed soundness

is sufficient in AC, range proofs with relaxed soundness do not suffice as drop-in replacement in AT (and their usage would lead to concrete attacks). Nevertheless, some (but not all) range proofs can be replaced with Sharp proofs in AT, and we sketch how Sharp proofs augmented with both a RSA and class group element improve this situation, even *without* trusted setup of the RSA modulus.

1.2. Technical Overview

1.2.1. CKLR Proofs.

Before introducing our technical improvements, we give a short overview of CKLR in the DLOG setting. Given a group $\mathbb{G}$ of order p with generators (G, H), a Pedersen commitment (Ped) to $x \in \mathbb{Z}_p$ with randomness r is given by $xG + rH$. (We use additive notation.)

CKLR opens the commitment to $x \in [0, B]$ in a zero-knowledge manner using standard Σ-protocol techniques. That is, the prover commits to random masks in $D = \mathsf{Ped.Commit}(\widetilde{x}, \widetilde{r})$, where $\widetilde{x}$ and $\widetilde{r}$ are additive masks for x and r respectively. Then, sends D to the verifier who in turn sends a random challenge $\gamma \in [0, \Gamma]$. The prover responds with two linear combinations $z = \gamma x + \widetilde{x}$, $t = \gamma r + \widetilde{r}$. Finally, the verifier checks the linear combination via $D + \gamma C = \mathsf{Ped.Commit}(z, t)$ and checks $z \in [0, (B\Gamma + 1)L]$, where L is the "masking overhead". We call such a "proof of opening with shortness check" a *proof of short opening (PoSO)*.

The basic observation in [Cou+21a] is that the soundness of the above protocol guarantees the extraction of a value of the form $x \equiv_p y \cdot \gamma^{-1}$, where both (y, γ) are short as well. While this does not suffice to bind the prover to a small integer, CKLR observes that $x \equiv_p y \cdot \gamma^{-1}$ uniquely defines a small rational number $u = y/\gamma \in \mathbb{Q}$ (where y, γ are short and coprime), if $2(B\Gamma + 1)\Gamma L \leq p$ holds.[2] We call $u \in \mathbb{Q}$ the *rational representative* of x and write $u = [x]_{\mathbb{Q}}$.

To show that u resides in the range $[0, B]$, CKLR decomposes $x(B - x) = \sum_{i \in [1,4]} y_i^2$ as the sum of four squares, commits to y_i in separate Ped commitments, performs a PoSO for the y_i and x, and shows that the decomposition holds over $\mathbb{Z}_p$ using the homomorphic properties of Ped. We call this part a *proof of decomposition (PoDec)*. The shortness guarantees of the PoSO imply that $u(B - u) \geq 0$ and thus $u \in [0, B]_{\mathbb{Q}}$, if $18((B\Gamma + 1)L)^2 \leq p$ holds.[3]

1.2.2. $\mathsf{Sharp}_{\mathsf{GS}}$: Group Switching and Batching via an Adapted PoDec.

To weaken the requirements on commitment homomorphism, we use a polynomial-based technique. That is, the prover commits to y_i in Ped commitments and performs a PoSO for each y_i, as before. To show that the four square decomposition holds, i.e. $x(B - x) = \sum_{i \in [1,4]} y_i^2$, the prover computes a polynomial f using the (short) masked witnesses $z = \gamma x + \widetilde{x}$ and $z_i = \gamma y_i + \widetilde{y}_i$ from the PoSO as follows:

$$f = z(\gamma B - z) - \sum_{i=1}^{4} z_i^2 = \alpha_2 \gamma^2 + \alpha_1 \gamma + \alpha_0.$$

A short computation shows that $\alpha_2 = 0$, i.e. the degree of f in γ is 1, iff the decomposition holds. To show that the degree of f is one, the prover commits to α_1 and α_0 in $C_* = \mathsf{Ped.Commit}(\alpha_1; r_*)$ and $D_* = \mathsf{Ped.Commit}(\alpha_0; \widetilde{r}_*)$ and sends C_*, D_* to the verifier. Then, the verifier sends the challenge γ and the prover replies with $t_* = \widetilde{r}_* + \gamma r_*$. Note that the verifier can recompute f from $z, \{z_i\}_{i=1}^4$ and the statement. Now, the verifier can check whether $f \equiv_q \alpha_1 \gamma + \alpha_0$ via

[2] CKLR interprets (y, γ, r) as a valid opening to u with respect to a modified Pedersen commitment that commits to rationals $u = y/\gamma$ as $(y \cdot \gamma^{-1})G + rH$ (or integers with rounding). Instead of relaxing the commitment, we relax the soundness guarantee of the range proof and keep working over rationals. This is more flexible and precise.

[3] For improved efficiency, CKLR and our protocols actually use a three square decomposition which can lead to problems in applications, see section 6.1.2. For simplicity, we stick with the four square decomposition in the introduction.

$\mathsf{Ped.Commit}(f, t_*) = D_* + \gamma C_*$. As the challenge is not known to the prover at the point of committing to the coefficients, the Schwartz–Zippel lemma guarantees that the decomposition holds over $\mathbb{Z}_q$ with overwhelming probability. Further, the prover reveals nothing about the values as the commitments are hiding and the openings are masked in t_*.

By construction, the polynomial-based technique allows us to use Pedersen multi-commitments (MPed), instead of separate Pedersen commitments (as in CKLR). Thus, we can perform N range proofs at once, with a constant number of group elements and a linear number of *short* integers.

The high level structure of this Σ-protocol resembles the lattice-based version of CKLR. But now, by committing to the entire decomposition y_i in a *single* Pedersen *multi*-commitment, which was not possible in the DLOG Σ-protocol of CKLR, the prover needs to communicate two integers and group elements fewer, compared to CKLR. This improves over the standard Σ-protocol for the showing the square decomposition in a group setting [CPP17; Cou+21a].

Group Switching. We highlighted in the overview above that the uniqueness of rational representatives requires (only) that $p \geq 2(B\Gamma + 1)\Gamma L$. Unfortunately, for the guarantee that the 3-square decomposition holds, this becomes $p \geq 18K^2$, where $K = (B\Gamma + 1)L$, which almost doubles the minimal possible group size. We observe that a dependency of PoSO and PoDec, which was present in CKLR, is removed with our improved Σ-protocol. Thus, we can choose groups with different modulus for the PoSO and PoDec. This gives us flexibility in group choices, and no compromise between optimal choice for commitment (typically 256-bit groups) or PoDec (typically larger groups) has to be made.

1.2.3. $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$: Cheaper Repetitions via a Novel PoSO

To clarify the requirements for our PoSO, we take a closer look at the security proof of $\mathsf{Sharp}_{\mathsf{GS}}$. The PoDec proves (among other equations) the square decomposition of N integers x_i:

$$x_i(B - x_i) \equiv_p \sum_{j=1}^{4} y_{i,j}^2 \tag{1.1}$$

for each committed value x_i. Security of PoDec follows from 3-special soundness, i.e. 3 related transcripts. To derive that $[x_i]_{\mathbb{Q}} \in [0, B]_{\mathbb{Q}}$, the security proof exploits a guarantee of the (simple) PoSO: Given two related transcripts $(a, \gamma, \vec{z})$ and $(a, \gamma', \vec{z}')$, we can extract $x_i \equiv_p \overline{z}_i/d$ where $\overline{z}_i = z'_i - z_i$ and $d = \gamma' - \gamma \in [-\Gamma, \Gamma]$, and likewise for $y_{i,j}$; given a third related transcript, eq. (1.1) is ensured. Moreover, $\overline{z}_i \in [-K, K]$ due to verifier size checks, so $[x_i]_{\mathbb{Q}} = \frac{\overline{z}_i}{d} \in \mathbb{Q}_{K,\Gamma}$, i.e. a fraction with numerator bounded by K and denominator bounded by Γ. Thus, multiplying eq. (1.1) by d^2, it is a homogeneous quadratic equation in d, B, $\overline{z}_i$, and $\overline{z}_{i,j}$, all of which bounded by K, so short. Since $18K^2 < p$, the equation holds over the integers. As a consequence, any PoSO which ensures that all extracted x_i, $y_{i,j}$ are of the form $x_i = \overline{z}_i/d$ and $y_{i,j} = \overline{z}_{i,j}/d$ is sufficient for this argument. Note that it is important that all fractions x_i, $y_{i,j}$ share the same denominator d for the above argument. Thus, we aim to replace the individual PoSOs by a "Batch-PoSO": Given any number of x_is (where we do not distinguish between x_i and $y_{i,j}$ anymore), prove that all of them are short fractions (i.e. in $\mathbb{Q}_{K,\Gamma}$) with a shared denominator d.

A straightforward approach is the following: To check shortness of $x_1, \ldots, x_N$, check shortness of the random linear combination $S = \sum_i \gamma_i x_i$ for $\gamma_i \leftarrow [0, \Gamma]$ (where we ignore masking terms for zero-knowledge for simplicity). Intuitively, if any x_i is not short,[4] the term $\gamma_i x_i$ should ensure that S is not short with high probability. And indeed, it is not hard to see that *individually*, every x_i is of the form $\overline{z}_i/d_i$ for short $\overline{z}_i$ and d_i, where $d_i \in [1, \Gamma]$. However, as we explained

[4]Recall that, e.g. $1/d \in \mathbb{Z}_p$, is considered short for $d \leq \Gamma$ in our setting.

above, we require that the *common denominator* d of all $\overline{z}_i/d_i$ is also short. Perhaps surprisingly, this does not follow trivially.

It is clear that, by using *binary* challenges, i.e. $\Gamma = 1$, all d_i are 1, and thus, the common denominator d is 1. In fact, all $\overline{z}_i/d_i = \overline{z}_i$ are small *integers*. This simple approach is well-known and used in (lattice-based) cryptography for proving knowledge of short preimages via random subset sums. While this even ensures standard soundness, it has the huge drawback of a binary challenge space. Thus, 128 repetitions are required for knowledge error 2^{-128}, which leads to relatively large proof size, e.g. instead of a 335-byte (relaxed sound) we get a 1877-byte (standard sound) range proof from $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ (for 32-bit range).

To achieve the claimed proof size, we must therefore choose a large challenge space $[0, \Gamma]$, so as to minimize repetitions. The crux of the security proof is then to ensure the common denominator d of all $\overline{z}_i/d_i$ is still short. Our core lemma (lemma 3.11) asserts, that either such a short common d exists, or the false acceptance probability at most $8/\Gamma$, This result is surprisingly non-trivial to prove, and it may be of independent interest.

Relation to similar lattice-based approaches As noted before, our Batch-PoSO bears close similarities to some (approximate) batch proofs of (knowledge of) short preimages in the lattice setting. Indeed, random linear combinations for batch proofs are a standard approach and used in the lattices setting, e.g. with binary challenges in [Bau+18a]. It is also used with larger challenges spaces to prove "fractional openings" of commitments, resulting in relaxed soundness somewhat similar to our setting, e.g. in [Ben+15; Bau+18b]. Namely, by multiplying with the (small) denominator, an extracted solution grows in size, but if parameters are chosen accordingly, the lattice problem still remains hard even for such larger solutions. Moreover, in special settings, e.g. ring-lattices, special challenge sets $\mathcal{C}$ where even $(\gamma' - \gamma)^{-1}$ is small for all $\gamma, \gamma' \in \mathcal{C}$ are used [AL21].

However, a crucial difference between our setting and the lattice-setting is that, in all the lattice-based works we are aware of, the challenge space for proving (approximate or relaxed) shortness is small and a large number of repetitions are required. Moreover, in these works, there is no requirement for a short *common* denominator d, instead, it suffices that *individually* each d_i is small, which is straightforward to show (but insufficient in our case). Since we embrace relaxed soundness and aim to maximize the challenge space, our approach exhibits such a requirement. Hence, to prove security, we require an entirely new analysis for the random linear combination test. Our current proof seems quite different from (advanced) lattice-based techniques, but it is an interesting question if and how such techniques are applicable to strengthen the lemma or simplify its proof.

Lastly, we note that lattice-based proof systems have vastly improved; even exact (range) proofs are now quite small, e.g. [LNS20; LNP22], though still an order of magnitude larger than group-based proofs, e.g. [LNP22] notes that a proof of opening alone needs 8 kB. We leave it as an interesting question, whether lattice-based range proofs could benefit from square-decompositions or our techniques as well.

1.2.4. $\mathsf{Sharp}_{\mathsf{HO}}$: Augmenting Sharp with Hidden Order Groups.

By using groups of hidden order, we can achieve improved soundness guarantees. On a high level, we add a single MPed commitment C' in a hidden order group to Sharp to restrict the possible commitment openings to "special" rationals. In contrast, all other range proofs in hidden order groups perform the *entire* range proof in the hidden order group [Bou00; Lip03; Gro05; CPP17; Cou+21a]. As these groups are larger than standard DLOG groups, our approach heavily improves efficiency.

Our proof of opening for the additional commitment only requires one additional short integer (for proving knowledge of the randomness of C'), as we use a *synthesized* challenge γ' and response

z'_i (computed from the actual challenges and responses) to avoid further repetitions (even if the underlying range proof is repeated). In more detail, when the PoSO is repeated R times with challenges $\{\gamma_k\}_{k=1}^R$, the prover and verifier set $\gamma' = \sum_{k=1}^R \gamma_k(\Gamma+1)^{k-1}$ and similarly for z'_i. So for completing the proof, only the masked commitment randomness t'_x is sent additionally. When instantiating this augmentation with suitable class groups, the committed x_is are restricted to be dyadic rationals, i.e. of the form $m/2^\ell$. With RSA groups, the x_i must be integers, hence the proof is standard sound.

2. Preliminaries

2.1. Notation and Basic Functions

We use log for the binary logarithm. We write $[a, b]$ for an interval $[a, b]$ in $\mathbb{Z}$, and we write $[a, b]_R$ for an interval in another space R, e.g. $\mathbb{Q}, \mathbb{R}, \mathbb{Z}_p$. We use Minkowski sum notation for sets, i.e. $A + B = \{a + b \mid a \in A, b \in B\}$ and write $A + b \coloneqq A + \{b\}$ for offsets. We denote by $|x|$ the absolute value of $x \in \mathbb{R}$. Let p be an (odd) (prime) number. Let $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$ be the integers modulo p, with representatives either $\mathbb{Z}_p = [0, p-1]$ or $\mathbb{Z}_p = [\lceil -\frac{p-1}{2} \rceil, \lceil \frac{p-1}{2} \rceil]$. Generally, we write $\equiv_p$ for equality $\mathsf{mod}\ p$ and $\in_{\mathbb{Z}_p}$ for set membership modulo p, i.e. $x \in_{\mathbb{Z}_p} S$ iff $\exists s \in S\colon x \equiv_p s$. For $x \in \mathbb{Z}_p$, let $|x| = \min\{|k| \mid k \in \mathbb{Z}, k \equiv_p x\} \leq p/2$. Recall that $d(x, y) = |y - x|$ for $x, y \in \mathbb{Z}_p$ defines a metric on $\mathbb{Z}_p$.

For a randomized algorithm $\mathscr{A}$ with input x, we write $y \leftarrow \mathscr{A}(x; r)$ for its execution with explicit randomness r. If the randomness is not explicit, we write $y \leftarrow \mathscr{A}(x)$ and assume that r was sampled accordingly. We write $s \xleftarrow{\$} S$ for sampling s uniformly at random from a finite set S or $d \xleftarrow{\$} D$ to sample d randomly according to a given probability distribution D. Further, we generally assume that some public parameters, denoted by pp, and the security parameter, denoted by λ, are implicitly passed as input to algorithms if it is clear by context.

We define the "prime number analogue" of the factorial.

Definition 2.1 (Primorial). We write $\mathsf{priml}(k)$ for the product of the first k primes, i.e. $\mathsf{priml}(k) \coloneqq \prod_{i=1}^k p_i$ where p_i is the i-th prime number.[5] We write $\mathsf{primlmin}(n)$ for $\min\{k \mid \mathsf{priml}(k) \geq n\}$, i.e. the smallest k such that $\mathsf{priml}(k) \geq n$.

2.2. Probability Theory

By U_X we denote the uniform distribution on a finite set X.

Definition 2.2. Let μ, ν be two probability measures on a *countable* set S. We define the **statistical distance** as

$$\Delta(\mu, \nu) = \sup_{A \subseteq S} \mu(A) - \nu(A) = \frac{1}{2} \sum_{a \in A} |\mu(\{a\}) - \nu(\{b\})|.$$

We define the **sup-ratio** $\rho(\mu/\nu)$ as

$$\rho(\mu/\nu) = \sup_{A \subseteq S} \mu(A)/\nu(A) = \sup_{s \in \mathrm{supp}(\mu)} \mu(s)/\nu(s)$$

where $0/0 = 1$ and $x/0 = \infty$ for $x > 0$.

Recall an important property of the sup-ratio: Given two random variables X and Y and any set of outcomes S, we have

$$\Pr[X \in S] \leq \rho(X/Y) \Pr[Y \in S].$$

[5] The usual definition of *primorial* is $n\# = \prod_{p_i \leq n} p_i$, where p_i is the i-th prime. That is, $n\#$ is the product of all primes p_i up to n. Thus, $\mathsf{priml}(k) = p_k\#$.

We will make ample use of this. Moreover, we use that

$$\rho((X', Y')/(X, Y)) \leq \rho(X'/X) \cdot \rho(Y'/Y)$$

for pair (X', Y') (resp. (X, Y)) of *independent* random variables.

Remark 2.3. Let $0 < d \leq C$ be integers and let $\gamma \xleftarrow{\$} [0, C-1]$ and $u' \xleftarrow{\$} [0, d-1]$. Suppose $u = \gamma \bmod d$. Then it is easily seen that $\rho(u/u') \leq 1 + d/C$. This follows, e.g., from $\rho(\gamma/\gamma') \leq 1 + d/C$ where $\gamma' \leftarrow [0, d\lceil C/d \rceil]$ and noting that $u' = \gamma' \bmod d$ in distribution.

2.3. Cryptographic Primitives

We define syntax and semantics of cryptographic primitives, and sketch their security properties. For formal definitions, see appendix A.

2.3.1. Cryptographic Groups.

We work in the DLOG setting with cryptographic groups. We write $\mathbb{G}$, $\mathbb{H}$, etc. for groups and use capital letters G, H, etc. for group elements. All groups are commutative and we use *additive* notation, i.e. we write $G + H$ and $x \cdot G$ or xG for $G, H \in \mathbb{G}$, $x \in \mathbb{Z}$. We denote by $\langle G \rangle$ the cyclic subgroup generated by G. The subgroup indistinguishability (SI) assumption in $\mathbb{G}$ asserts that $H \xleftarrow{\$} \mathbb{G}$ and $H \xleftarrow{\$} \langle G \rangle$ are indistinguishable.

A PPT algorithm GenGrp on input 1^λ outputs a (description of a) group $\mathbb{G} = \mathbb{G}_\lambda$. Given the description, group operations (addition and inverse) and membership tests are efficient, as well as bounds $U_{\mathsf{lo}} \leq |\mathbb{G}| \leq U_{\mathsf{up}}$ on the group order are specified. For notational simplicity, we leave GenGrp implicit in the rest of the work. By $A \xleftarrow{\$} \mathbb{G}$ we denote randomly drawn group elements *without* trapdoors.[6] When we say "$\mathbb{G}$ is a group of (prime) order $p = p_\lambda$", we mean that $p = |G|$ is known unless explicitly stated otherwise.

The DLOG assumption in cyclic groups asserts that finding the discrete logarithm of a random group element $H \xleftarrow{\$} \mathbb{G}$ is hard. It translates to groups of hidden order (where $\langle G \rangle \subsetneq \mathbb{G}$ is possible), by considering $H \xleftarrow{\$} \langle G \rangle$. For better efficiency in groups of large (possibly unknown) order, the DLOG assumption can be strengthened.

Definition 2.4 (DLSE, SEI). The **S-bounded DLSE assumption** asserts that it is hard to compute DLOG (w.r.t. G) of zG where $z \xleftarrow{\$} [0, S]$. The **$S$-bounded SEI assumption** asserts that it is hard to distinguish (G, H) and (G, H') where $H \xleftarrow{\$} \langle G \rangle$ and $H' = zG$ for $z \xleftarrow{\$} [0, S]$ (and $G \xleftarrow{\$} \mathbb{G}$).

The above assumptions are only of interest if $S \ll \mathsf{ord}(\mathbb{G})$. Throughout this work, we generally set $S = 2^{2\lambda} - 1$.[7]

2.3.2. Hash Functions.

A (keyed) hash function Hash is of the form $\mathsf{Hash} \colon \mathcal{K} \times \{0,1\}^* \mapsto \{0,1\}^\ell$. The key (i.e. the first input) to Hash is usually implicit, and part of the public parameters. We call Hash a **collision-resistant** hash function (CRHF), if it is hard to find a collision, i.e. two inputs m, m' such that $\mathsf{Hash}(m) = \mathsf{Hash}(m')$.

[6] Transparent setup typically requires trapdoor-free sampling. Otherwise, A could be sampled/encoded via $x \in \mathbb{Z}$, as $A = xG$, leaking the dlog of A. A stronger form, called *invertible sampling* is often used in security reductions to "program" the setup, and possible in most cryptographic groups (including $\mathbb{Z}_p^\times$, elliptic curves, and RSA groups). However, as noted in [Abr+22], there are no known invertible sampling algorithms for class groups. In this work, we rely on suitably strengthened hardness assumptions to avoid invertible sampling in class groups.

[7] To the best of our knowledge, there are no non-generic attacks on the (short) discrete logarithm assumption in hidden order groups. The best generic algorithm (without preprocessing) has $\mathcal{O}(\sqrt{S})$ runtime, see for example [CK18, Section 3.2].

2.3.3. Commitment Schemes.

A (non-interactive) **commitment scheme** Com allow committing to a message m, obtaining a commitment c and opening information d. More formally, Com is a 3-tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$ s.t.

- $\mathsf{Com.Setup}(1^\lambda)$: outputs a commitment key ck (often left implicit),
- $\mathsf{Com.Commit}_{\mathrm{ck}}(x)$: outputs a pair (c, d) of commitment c (to x) and opening d under commitment key ck,
- $\mathsf{Com.Verify}_{\mathrm{ck}}(c, x, d)$: outputs 1 iff it accepts that c opens to x given opening d under commitment key ck.

We require that Com is (perfectly) **correct**, i.e. honest commitments always verify. Moreover, Com should be **binding** and **hiding**, i.e. a commitment c can be opened to (at most) one message x, and it is hard to distinguish whether an (unopened) commitment is to message x_0 or x_1.

Instantiation. We consider Pedersen multi-commitments (MPed), a generalization of the Pedersen commitment scheme [Ped92], with short openings over a prime or hidden order group $\mathbb{G}$. Let $N, S \in \mathbb{N}$ and $U_{\mathsf{lo}} \leq |\mathbb{G}| \leq U_{\mathsf{up}}$. Setup samples $G_i \xleftarrow{\$} \mathbb{G}$ for $i \in [0, N]$ and outputs commitment key $\mathrm{ck} = (\{G_i\}_{i\in[0,N]})$. Given a message vector $\{x_i\}_{i\in[1,N]}$, Commit samples $r \xleftarrow{\$} [0, S]$, sets $C = rG_0 + \sum_{i\in[1,N]} x_i G_i$, and outputs the pair (C, r). Given commitment C, message $\{x_i\}_{i\in[1,N]}$ and opening r, Verify outputs 1 iff $C = rG_0 + \sum_{i\in[1,N]} x_i G_i$ and x_i is in the right message space for all i. That is, if $\mathbb{G}$ has prime order p, then $x_i \in \mathbb{Z}_p$, or else $x_i \in \mathbb{Z}$ unless stated otherwise. We write Ped for the Pedersen commitment scheme, i.e. MPed for $N = 1$. The scheme MPed is hiding under the SI and SEI assumptions and binding under the DLOG assumption. The strength of the hiding property scales with hiding parameter S.[8]

2.3.4. Zero-Knowledge Proofs of Knowledge.

A proof system (P, V) for NP-relation R is a two-party protocol, where prover P has input $(x, w) \in \mathsf{R}$ and verifier V has input x. The verifier accepts or rejects an interaction (by outputting 1 or 0). The prover has no output. Moreover, we require correctness with error γ_{err}, that is if $(x, w) \in \mathsf{R}$, then in an honest execution, the verifier accepts except with probability γ_{err}.

Our proof systems will be **proofs of knowledge** (PoK) and **non-abort special honest verifier zero-knowledge** (SHVZK). PoK means, that one can extract a witness w for x from any prover which convinces V with probability higher than the knowledge error κ_{err}. We consider **relaxed soundness**, that is, the witness relation $\mathsf{R}_{\mathsf{Ext}}$ for an extracted witness can differ from the correctness relation R. We share this efficiency trade-off with many lattice-based proof systems. Non-abort SHVZK means, that transcripts where the prover does not abort can be simulated efficiently given only x, if the verifier's challenges are known ahead of time. In our proof systems, prover aborts happen due to rejection sampling.

We work in the **common reference string (CRS)** model. Most of our protocols require only a **uniform (common) random string (URS)**, a.k.a. transparent setup.

[8] If $G_i \xleftarrow{\$} \langle G_0 \rangle$ and S is large enough, then MPed is statistically hiding. Under the SI assumption, instead using $G_i \xleftarrow{\$} \mathbb{G}$ remains (computationally) hiding. Usually, sampling $G_i \xleftarrow{\$} \mathbb{G}$ can be transparent (trapdoor-free), but $G_i \xleftarrow{\$} \langle G_0 \rangle$ not necessarily.

2.3.5. Random Oracle Model (ROM)

In the ROM, all parties have access to a truly random function $\mathsf{RO}\colon \{0,1\}^* \to \{0,1\}^{2\lambda}$. The Fiat–Shamir transformation converts public coin protocols to non-interactive zero-knowledge proofs of knowledge (NIZKPoK) by computing the verifier's challenges as hashes over partial transcripts and other context information (which includes x). In case of non-zero correctness error, one retries in case of aborts [Lyu09]. In practice, the ROM is heuristically instantiated by a strong cryptographic hash function, e.g. SHA-3. Note that a URS can be generated trivially in the ROM.

2.4. Rational Representatives

Using $\mathbb{Z}$-valued representatives for $\mathbb{Z}/p\mathbb{Z}$ is a natural choice, obtained from the homomorphism $\mathbb{Z} \to \mathbb{Z}_p$, $x \mapsto x \bmod p$. Another choice is induced by the ring $\mathbb{Z}_{(p)} = \{\frac{n}{d} \mid n \in \mathbb{Z}, d \in \mathbb{N}, p \nmid d\} \subseteq \mathbb{Q}$, and the homomorphism $\frac{n}{d} \mapsto n \cdot (d^{-1} \bmod p) \bmod p$. We call such representatives *rational*. Strictly speaking, a set of representatives $R \subseteq \mathbb{Z}_{(p)}$ should have a *unique* representative for each element in $\mathbb{Z}_p$. We work with smaller sets, which do not have representatives for all of $\mathbb{Z}_p$, but existing representatives are unique. The lack of surjectivity will be of no concern since, by construction, elements of interest will always come with an admissible representative.

Definition 2.5. Let $\mathbb{Q}_{N,D} \subseteq \mathbb{Q}$ be the rationals whose numerator is bounded by N and denominator bounded by D, that is

$$\mathbb{Q}_{N,D} = \{\frac{n}{d} \in \mathbb{Q} \mid |n| \leq N, |d| \leq D\} \subseteq \mathbb{Q}.$$

The value x is represented by $\frac{n}{d}$ if $x \equiv_p nd^{-1}$ (where d^{-1} is computed modulo p).

Note that we interpret $\frac{n}{d}$ as a fraction; the tuple (n, d) is not unique. It becomes unique if $\frac{n}{d}$ is reduced and $d \geq 1$.

Lemma 2.6 (Criterion for Unique Representative in $\mathbb{Q}_{N,D}$)**.** *Let N, D so that $N \cdot D < p/2$. Then for any $x \in \mathbb{Z}_p$, if there is a representative in $\mathbb{Q}_{N,D}$ of x, i.e. some $\frac{n}{d}$ so that $nd^{-1} \equiv_p x$, then $\frac{n}{d}$ is unique (as a fraction).*

Proof. Suppose $x \equiv n_i d_i^{-1}$ (p) for $i = 1, 2$. Then $n_1 d_2 \equiv n_2 d_1$ (p). Since $N \cdot D \leq p/2$ and $\frac{n_i}{d_i} \in \mathbb{Q}_{N,D}$, we find that $n_1 d_2 = n_2 d_1$ over $\mathbb{Z}$. (No wrap-around.) Thus, $\frac{n_1}{d_1} = \frac{n_2}{d_2}$ as fractions, and the claim follows. □

From now on, we always assume that $N \cdot D < p/2$ whenever we use $\mathbb{Q}$-representatives.

Remark 2.7. Let $a \in \mathbb{Z}_p$ and $ND < p/2$. We define $[a]_{\mathbb{Q}} \in \mathbb{Q}_{N,D}$ as the unique irreducible representatives $\frac{n}{d}$ of a, assuming it exists. (We assume that some maximal bounds N, D are implicitly fixed in the context.) We note that $[a]_{\mathbb{Q}}$ can be efficiently computed (if it exists), see [FSW03].

2.5. Masking Scheme

We use "additive masking" to hide information with random noise. For readability, we use an abstraction of this technique formalized below, in a way similar to [ACK21]. A **masking scheme** is a tuple $(\mathsf{R}, \mathsf{mask}, V)$ of efficiently samplable distribution R and a masking algorithm mask for values in range $[0, V]$.

- $r \xleftarrow{\$} \mathsf{R}$ is an integer $r \in [0, (V+1)L]$, i.e. $\mathrm{supp}(\mathsf{R}) \subseteq [0, (V+1)L]$. We call r the *mask* and $L \geq 1$ the **masking overhead**.
- $\mathsf{mask}(v, r)$ takes as input an integer $v \in [0, V]$ and a mask r and outputs $v + r$ or $\perp$. For simplicity, we require $\mathsf{mask}(v, r) = \perp$ if $v + r \notin [0, (V+1)L]$.

- p denotes an upper bound on the **abort probability**, that is, a bound satisfying $\sup_{v\in[0,V]} \Pr[\,\mathsf{mask}(v,r) = \perp \mid r \xleftarrow{\$} \mathsf{R}\,] \leq \mathsf{p}$.
- Let M_v denote the distribution defined via: Sample $r \xleftarrow{\$} \mathsf{R}$, then return $\mathsf{mask}(v,r)$. Then $\varepsilon_{\mathsf{mask}} = \sup_{v,w\in[0,V]} \Delta(M_v, M_w)$ is called the **masking error**.

The range V is sometimes left implicit. Intuitively, $z = \mathsf{mask}(v,r)$ reveals almost nothing about v, since the random mask r ensures that z is distributed (almost) independently from v. The masking error quantifies this intuition.

Rejection Sampling. (Uniform) Rejection sampling is usually described for values in intervals $[-V,V]$, i.e. symmetric around 0. We use $[0,V]$ instead, and adapt mask accordingly. Namely, for given masking overhead L:

- The distribution R is the uniform distribution $U_{[0,(V+1)L]}$.
- $\mathsf{mask}(v,r)$ outputs $v+r$ if $v+r \in [V,(V+1)L]$, else $\perp$.
- The abort probability is $\mathsf{p} = \frac{V+1}{(V+1)L+1} \leq \frac{1}{L}$.[9]
- The masking error is 0. [10]

Drowning in noise. In the above, set $L = 2^\lambda$. Then abort probability is $2^{-\lambda}$. This is convenient to use if "size" of r does not matter much.

No aborts. We also use masking schemes to save communication. In these cases, once R grows beyond $\mathbb{Z}_p$, i.e. $\mathbb{Z}_p = [0,p-1] \subseteq \mathsf{R}$, we assume that $\mathsf{R} = \mathbb{Z}_p$ and $\mathsf{mask}(v,r) = v + r \bmod p$ (without abort). We will be explicit about such potential optimizations.

3. Shortness Testing mod p

In this section, we present a result that allows us to test shortness of many fractions at once. We will apply this result later to efficiently test shortness of committed values in our range proofs (see section 5). Indeed, it is the basis for constructing a range proof which communicates a *single* integer per repetition. For readability, we only present proof sketches and sometimes simplified claims. The full claims and proofs can proofs are in appendix D. First, we define a notion of "shortness test" which is tailored to our application.

Definition 3.1 (Fractional Shortness Test). A *(fractional) shortness test* is an algorithm T which takes as input $\vec{x} \in \mathbb{Z}_p^N$ (where T is implicitly parameterized by p and N) and outputs $T(\vec{x}) \in \{0,1\}$. Let $K, D \in \mathbb{N}$ with $KD < p/2$. A vector $\vec{x} \in \mathbb{Z}_p^N$ is *uniformly* (K,D)*-short*, if $\exists d \in [1,D] \colon d\vec{x} \in [-K,K]^N_{\mathbb{Z}_p}$. Let $\phi_{K,D}(\vec{x}) \in \{0,1\}$ be the predicate which is 1 if $\vec{x}$ is uniformly (K,D)-short. We say that T is a *fractionally* (K,D)*-sound* shortness test with error κ, if

$$\forall \vec{x} \in \mathbb{Z}_p^N \colon \quad \Pr[T(\vec{x}) = 1 \implies \phi_{K,D}(\vec{x}) = 1] \geq 1 - \kappa \tag{3.1}$$

or, equivalently,

$$\forall \vec{x} \in \mathbb{Z}_p^N \colon \phi_{K,D}(\vec{x}) = 0 \implies \Pr[T(\vec{x}) = 1] \leq \kappa. \tag{3.2}$$

[9] For any $v \in [0,V]$, there are $V+1$ "bad" r (out of $(V+1)L+1$ choices for r).
[10] The abort probability is independent of v. Conditioned on no abort, the distribution is uniform over $[V,(V+1)L]$.

The crucial point in fractional (K, D)-soundness is that a vector is rejected with high probability if there is no *single* denominator of size at most D such that $d \cdot \vec{x} \in [-K, K]^N_{\mathbb{Z}_p}$, i.e. $\|d \cdot \vec{x}\|_\infty \leq K$. A weaker definition might only require $x_i \in \mathbb{Q}_{K,D}$ for all i, but this is not enough for our applications. Note that we do not define what correctness of a fractional shortness test is; it will be evident in applications and concrete requirements may vary.

Definition 3.2 (RAST). We define the *random affine shortness test* $\mathsf{RAST}_{N,\mathcal{D},K,\mu}$ for shortness over $\mathbb{Z}_p$ with *dimension* or *batch-size* N, *test distribution* $\mathcal{D}_N$ *range bound* K, and *offset* μ as follows: To test $\vec{x} \in \mathbb{Z}_p^N$, pick $\vec{\gamma} \xleftarrow{\$} \mathcal{D}_N$, and output 1 if $\mu + \sum_{i=1}^N x_i\gamma_i \in [0, K]_{\mathbb{Z}_p}$, else output 0.

The following theorem assures fractional soundness of the RAST. The proof is based on the core lemma, lemma 3.11, whose proof is technical and lengthy; it is the subject appendix D.

Theorem 3.3. *Let* RAST *be the random affine shortness test with uniform distribution* $\mathcal{D}$ *over* $[0, D]^N$*, dimension* N*, range bound* K*, and any offset* $\mu \in \mathbb{Z}_p$*. Let* $K' = (1 + 2\beta)K$ *where* $\beta = \min(N, \mathsf{primlmin}(D+1))$ *and suppose that* $2D(K' + DK + 2) < p$*. Then* RAST *is fractionally* (K', D)*-sound with error* $8/(D + 1)$*,*

content/proof-shortness-test-soundness

3.1. Modulo Arithmetic

In this section, we work with representatives $\mathbb{Z}_p = [0, p - 1]$ instead of representatives which are symmetric around 0. Mostly, because we want to use remark 3.4. However, our results are phrased in a way which is independent of representatives, so they hold for any choice of representatives for $\mathbb{Z}_p$.

First, recall that a (rational) number x splits into an integer part $\lfloor x \rfloor$ and a decimal part $x - \lfloor x \rfloor$, often denoted $\mathrm{frac}(x)$.

Remark 3.4. For $m \in \mathbb{Z}$, $d \in \mathbb{N}$, we make much use of following simple but important equality:

$$\frac{m}{d} = \left\lfloor \frac{m}{d} \right\rfloor + \frac{m \bmod d}{d} = \left\lceil \frac{m}{d} \right\rceil - \frac{m \bmod d}{d}. \tag{3.3}$$

This equality holds for representatives $[0, d - 1]$ of $\mathbb{Z}_d$ for "$x \bmod p$". For modulo operations symmetric around 0, "flooring"/"ceiling" would become "rounding".

Remark 3.5 (Inequalities for floor and ceil). Let $x, y \in \mathbb{R}$. The we have $\lfloor x \rfloor \leq x \leq \lceil x \rceil$ and

$$\lfloor x \rfloor + \lfloor y \rfloor \leq \lfloor x + y \rfloor \leq x + y \leq \lceil x + y \rceil \leq \lceil x \rceil + \lceil y \rceil \tag{3.4}$$

Lemma 3.6 (Regular Spacing of $\mathbb{S}_d$)**.** *Suppose* $1 < d < p$ *and* $\gcd(d, p) = 1$ *and consider the set*

$$\mathbb{S}_d \equiv_p \{\frac{i}{d} \bmod p \mid i \in [0, \ldots, d - 1]\} \subseteq \mathbb{Z}_p. \tag{3.5}$$

Then $\mathbb{S}_d = \{\lceil ip/d \rceil \mid i \in [0, \ldots, d - 1]\}$ *and the minimal distance* $\delta = \min_{x \neq y \in \mathbb{S}_d} |x - y|$ *satisfies* $\delta = \lfloor \frac{p}{d} \rfloor$.

When interpreting $\mathbb{Z}_p$ and the set S on the unit circle as regularly spaced points, it is visually clear that the claim should hold. See fig. 1 for this. Note, $d/d \equiv_p 1$, that is, it is an angle of $2\pi/p$ away from 0, so the spacing of $\mathbb{S}_d$ is not perfectly regular. Indeed, as shown in fig. 1, the points $i/d \in \mathbb{Z}_p$ are not in sequential order, but permuted by a unit modulo d, so the visual heuristics can be somewhat misleading. With lemma 3.6 at hand, we can easily derive some simple consequences.

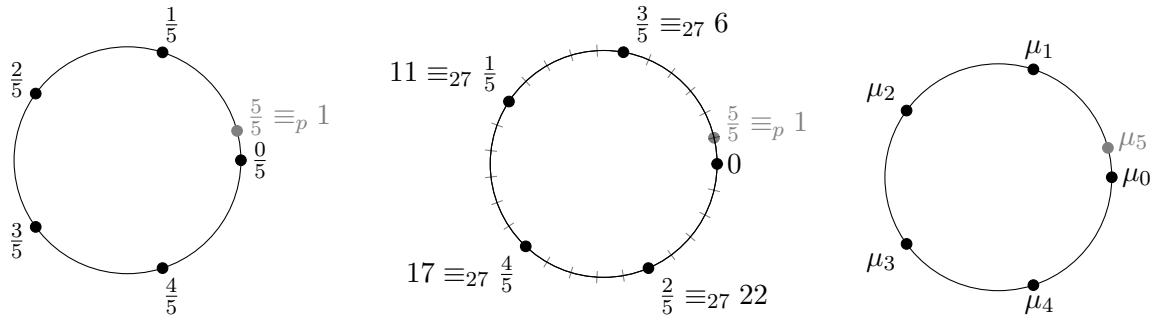

Figure 1: *Visual heuristics for lemma 3.6.* The left figure is the naive intuition. The middle figure is the visualization of for $p = 27$. The right figure denotes $\mathbb{S}_d = \{\mu_0, \ldots, \mu_4\}$ where $\mu_i \equiv_p \lceil \frac{ip}{d} \rceil$. In the example, $\mu_1 \equiv_{27} 6$, $\mu_2 \equiv_{27} 11$, $\mu_3 \equiv_{27} 17$, $\mu_4 \equiv_{27} 22$.

Lemma 3.7. *Suppose $d \in \mathbb{N}$ and $\gcd(d, p) = 1$ and $u \xleftarrow{\$} [0, \ldots, d-1]$. Let $\mu, K \in \mathbb{N}$ be arbitrary. Then for $1 < d < p$ we have*

$$\Pr\left[\frac{u}{d} \in_{\mathbb{Z}_p} [0, K]_{\mathbb{Z}_p} + \mu\right] \leq \frac{1}{d}\left\lceil \frac{K+1}{\lfloor \frac{p}{d} \rfloor} \right\rceil \tag{3.6}$$

and for $d > p$, we have

$$\Pr\left[\frac{u}{d} \in_{\mathbb{Z}_p} [0, K]_{\mathbb{Z}_p} + \mu\right] \leq 2\frac{K+1}{p} \tag{3.7}$$

where the probability is over u. Combining the conditions gives

$$\Pr\left[\frac{u}{d} \in_{\mathbb{Z}_p} [0, K]_{\mathbb{Z}_p} + \mu\right] \leq \frac{1}{d} + 2\frac{K+1}{p} \tag{3.8}$$

Note that in lemma 3.7, we consider membership intervals $[0, K] + \mu$, i.e. arbitrary (shifted) intervals, not just $[0, K]$, because such intervals appear naturally in our setting.

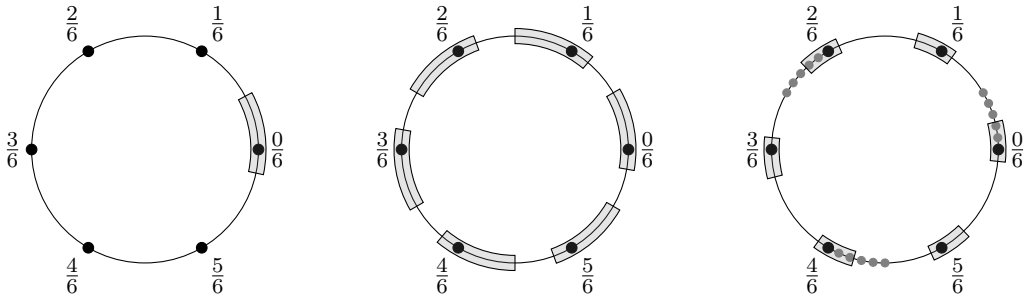

Figure 2: *Visual heuristics for lemmata 3.7 and 3.8.* The left figure is the intuition for lemma 3.7. The middle figure shows $[0, K]_{\mathbb{Z}_p} + \mathbb{S}_6 + \mu$. The right figure shows $[0, K]_{\mathbb{Z}_p} + \mathbb{S}_6 + \mu$ and how points of the form $ua/3$ are distributed. Visibly, until the gray points first escape the intervals, they all lie within; after they escape, they never re-enter an interval. Hence at most $3 \cdot K/a$ points lie within $[0, K]_{\mathbb{Z}_p} + \mathbb{S}_6 + \mu$.

While lemma 3.7 was a good warm-up, it does not cover all of our needs. On the one hand, we need to deal with more general a/b (instead of just $1/d$) and $u \in [0, D]$ (instead if $u \in [0, d-1]$). On the other hand, we need to deal with unions of disjoint intervals, namely $[0, K] + \mathbb{S}_d + \mu$. Thus, we take a closer look at the specific case of interest. It is sketched in fig. 2.

Lemma 3.8. *Let $p, d, a, b, \mu, D, K \in \mathbb{N}$ and suppose $u \xleftarrow{\$} [0, D]$ is a uniform random variable. Let $\mathbb{S}_d \equiv_p \{i/d \mid i \in \mathbb{Z}_d\} \subseteq \mathbb{Z}_p$ as usual, and likewise $\mathbb{S}_b$. Suppose that $\gcd(d, p) = 1$, and $b \mid d$, and that*

$$b(K+1) + Da < \left\lfloor \frac{p}{d/b} \right\rfloor. \tag{3.9}$$

Then we have

$$\sum_{s \in \mathbb{S}_d} \Pr\left[u\frac{a}{b} \in_{\mathbb{Z}_p} [0, K]_{\mathbb{Z}_p} + \mu + s\right] \leq \left\lceil \frac{b(K+1)}{a} \right\rceil \frac{1}{D+1}. \tag{3.10}$$

Again, the claim of lemma 3.8 is visually clear and sketched in fig. 2. (Our lemma is not as tight as the picture suggests, but good enough for our purposes.)

Remark 3.9. A useful property of $\mathbb{S}_d$ is that $\mathbb{S}_d = 1 - \mathbb{S}_d$ (because $\frac{p-i}{d} \equiv_p 1 - \frac{i}{d}$), hence, lemma 3.8 also applies to $[0, K] + \mu - \mathbb{S}_d \equiv_p [0, K] + (\mu - 1) + \mathbb{S}_d$. Moreover, lemma 3.8 applies to negative a as well, where $|a|$ is used in all bounds and estimates. This follows since multiplying the expression in the probability by (-1) leads to positive n which must lie in $[-K, 0]_{\mathbb{Z}_p} - \mu - \mathbb{S}_d$ which can be rewritten as $[0, K]_{\mathbb{Z}_p} + \mu' + \mathbb{S}_d$ for suitable μ'.

Towards analyzing random linear combinations with the help of lemmata 3.7 and 3.8, we introduce another lemma.

Lemma 3.10 (Simplified Lemma D.2)**.** *Suppose $1 \neq d \in \mathbb{N}$ and let u_i be random variables in $\mathbb{Z}_d = [0, \ldots, d-1]$ for $i = 1, \ldots, N$. Fix some arbitrary $a_i \in [0, d-1]$ with $d = \operatorname{lcm}(a_1, \ldots, a_N)$. Then there exist $q_1, \ldots, q_N \in \mathbb{N}$, which are pairwise coprime, $q_i \mid \mathsf{ord}_{\mathbb{Z}_d}(a_i)$, and $\prod_{i=1}^N q_i = d$. Let $Z = \prod_{i=1}^N \mathbb{Z}_{q_i} \hookrightarrow \mathbb{Z}_d^N$ (where the injections $\mathbb{Z}_{q_i} \hookrightarrow \mathbb{Z}_d$ of the Chinese remainder theorem is used component-wise). Then $\sum_{i=1}^N u_i \cdot a_i \bmod d$ is uniformly distributed in $\mathbb{Z}_d$ for $(u_1, \ldots, u_N) \xleftarrow{\$} Z$.*

Clearly, in lemma 3.10, $\sum_{i=1}^N u_i a_i$ is uniformly distributed if $u_i \xleftarrow{\$} \mathbb{Z}_d$ for all i. The key point of lemma 3.10 is, that the sum is uniformly distributed even if the u_i are drawn from the possibly much smaller space Z. This helps in our analysis of the core lemma.

3.2. Shortness Failure of Random Linear Combinations

Now, we turn to the core lemma, lemma 3.11. It should be viewed as a non-trivial generalization of lemmata 3.7 and 3.8 with certain requirements and restrictions. Implicit in lemma 3.11 is the RAST from theorem 3.3. That is, we consider the probability of "bad" challenges, which for a given choice of x_i's of the form $\frac{m_i}{d_i}$ lead to falsely accepting $\sum_i \gamma_i x_i$ as short, even though some x_i exceed the allowed bounds.

Lemma 3.11 (Core Lemma)**.** *Let $D, M \in \mathbb{N}$ and suppose $2DM < p$. Let $x_i = \frac{m_i}{d_i}$ where $d_i \in [1, D]$ and $m_i \in [-M, M]$ for $i = 1, \ldots, N$. Let $\gamma_i \xleftarrow{\$} [0, D]$. Define*

$$S = \sum_{i=1}^{N} \gamma_i \cdot \frac{m_i}{d_i} \bmod p \tag{3.11}$$

Let $I \subseteq [1, N]$ denote the set of indices which minimizes $d \coloneqq \operatorname{lcm}(\{d_i \mid i \in I\})$ under the constraint that $d > D$, or $I = [1, N]$ if $\operatorname{lcm}(d_1, \ldots, d_N) \leq D$. Let $K \in \mathbb{N}$, let $\beta = \min(|I|, \mathsf{primlmin}(D+1))$, and let $K' \coloneqq K + 2\beta M$. Then, for arbitrary $\mu \in \mathbb{Z}_p$, we have

$$\Pr[S \in [0, K]_{\mathbb{Z}_p} + \mu] \leq 4 \cdot \begin{cases} \frac{1}{d} & \text{if } d(K'+1) < p \\ \frac{1}{d} + 2\frac{K'+1}{p} & \text{always} \end{cases} \tag{3.12}$$

Now, suppose additionally that $d \leq D$ and $D(K' + DM + 2) < p$. If $\frac{d}{d_i}|m_i| > K'$ for some $i \in [1, N]$, then

$$\Pr[S \in [0, K]_{\mathbb{Z}_p} + \mu] \leq \frac{8}{D+1}. \tag{3.13}$$

This time, we have no "visual heuristic". However, though the detailed proof of lemma 3.11 is rather technical, its basic idea is relatively simple: First, simplify the situation by reducing to the case of $I = \{1, \dots, N\}$ and imposing certain minimality properties on I and d. Then, rewrite the sum S with lowest common denominator d. That is, let $S' = \sum_{i=1}^{N} \gamma_i \cdot \frac{dm_i}{d_i} \in \mathbb{Z}$, and then split S'/d (resp. S) into decimal and integer parts:

$$S \equiv_p \frac{1}{d} S' = \frac{S' \bmod d}{d} + \left\lfloor \frac{S'}{d} \right\rfloor.$$

The idea is to exploit this to analyze the two summands separately, after making them *almost* stochastically independent. (But this works only to some extent, namely, small garbage terms will appear.) For this, we change the challenge distribution. For γ_i, we could change $U_{[0,D]}$ to $U_{[0,d_i\lceil \frac{C+1}{d_i} \rceil]}$, which allows us to write $\gamma_i' = u_i + d_i v_i$ with $u_i \xleftarrow{\$} [0, d_i - 1]$ and $v_i \xleftarrow{\$} [0, \lceil \frac{C+1}{d_i} \rceil]$. Then

$$S'/d = \sum_{i=1}^{N} \gamma_i' \frac{m_i}{d_i} = \sum_{i=1}^{N} u_i \frac{m_i}{d_i} + \sum_{i=1}^{N} v_i m_i.$$

On the right hand side, the second sum is an integer sum, and relatively easy to control. The first sum has the same form as S', but the u_i are uniformly from $\mathbb{Z}_{d_i}$ now, which is simpler to analyze. However, our analysis makes use of lemma D.2 to get a tighter result. Lemma D.2 suggests $\gamma_i' \sim U_{[0,q_i\lceil \frac{C+1}{q_i} \rceil]}$ (for suitable q_i), and we write

$$S \equiv_p \frac{1}{d} S_u + S_v \quad \text{where} \quad S_u = \sum_i u_i \frac{m_i d}{d_i} \quad \text{and} \quad S_v = \sum_i v_i q_i \frac{m_i}{d_i}.$$

The central requirements of this change in distribution are that it is close (in terms of $\rho(\vec{\gamma}/\vec{\gamma}')$), that $S_u \bmod d$ is now uniformly distributed, and that S_u and S_v are stochastically independent. Indeed, a "loss factor" of 4 compared to lemma 3.7 comes precisely from $\rho(\vec{\gamma}'/\vec{\gamma})$. Moreover, when rewriting $\frac{1}{d} S_u = \frac{S_u \bmod d}{d} + \lfloor \frac{S_u}{d} \rfloor$, we can get rid of the garbage term $\lfloor \frac{S_u}{d} \rfloor = \sum_i u_i m_i \frac{q_i}{d_i}$ by increasing the interval from $[0, K]$ to $[0, K + 2\beta M]$ (since the garbage term lies in $[-\beta M, \beta M]$). After these changes, we have simplified to

$$\begin{aligned} &\Pr[S \in_{\mathbb{Z}_p} [0, K]_{\mathbb{Z}_p} + \mu] \leq \\ &\rho(\vec{\gamma}/\vec{\gamma}') \cdot \Pr\left[\frac{S_u \bmod d}{d} + S_v \in_{\mathbb{Z}_p} [0, K + 2\beta M]_{\mathbb{Z}_p} + \mu'\right] \end{aligned}$$

where S_u and S_v are independent and $(S_u \bmod d)$ is uniform in $\mathbb{Z}_d$. Now, lemma 3.11 follows effectively from lemma 3.7 (which yields eq. (3.12)) and lemma 3.8 (which yields eq. (3.13)).

The core lemma is precise enough for our purposes, but the true bounds and premises may be much better. On the one hand, the necessity of the size restrictions on D, K, M is uncertain, as is the role of β. On the other hand, a factor of 4 in the inequalities in lemma 3.11 is a consequence of switching $\vec{\gamma}$ to $\vec{\gamma}'$ (and relying on lemma D.2 to "shrink" the randomness space) instead of analyzing the distribution of the sum S more directly.

4. $\mathsf{Sharp}_{\mathsf{GS}}$: Batching and Group Switching

In this section, we present the optimized Σ-protocol for showing the decomposition in the DLOG setting, introduce group switching, and show how to perform efficient proofs for batches of integers.

4.1. Parameters

Here, we give an overview of all the used parameters in $\mathsf{Sharp}_{\mathsf{GS}}$. Let $N \in \mathbb{N}$ be the number of integers $x_1, \ldots, x_N$ in the ranges $[0, B_i]$. In the following, we fix $B = B_i$ for simplicity. Let R be the number of repetitions of the proof and $[0, \Gamma]$ be the challenge set. Generally, we have $R = \lceil \lambda / \log(\Gamma + 1) \rceil$ unless lower soundness than λ bits is satisfactory. We will need to mask values $x \in [0, B\Gamma]$ and values $r \in [0, S\Gamma]$ (where S is defined below) with masking algorithm $\mathsf{mask}_x, \mathsf{mask}_r$, masking randomness distribution R_x, R_r, masking overhead L_x, L_r and masking abort probability $\mathsf{p}_\mathsf{x}, \mathsf{p}_\mathsf{r}$ respectively. Let $p \geq 2(B\Gamma^2 + 1)L_x$ and $q \geq 18((B\Gamma + 1)L_x)^2$. We use MPed commitments with hiding parameter S in groups $\mathbb{G}_{\mathsf{com}}$ and $\mathbb{G}_{\mathsf{3sq}}$, with prime order p and q respectively. We fix generators $G_0, G_i, G_{i,j} \xleftarrow{\$} \mathbb{G}_{\mathsf{com}}$ for the commitment key $\mathrm{ck}_{\mathbb{G}_{\mathsf{com}}}$ and $H_0, H_i \xleftarrow{\$} \mathbb{G}_{\mathsf{3sq}}$ for $\mathrm{ck}_{\mathbb{G}_{\mathsf{3sq}}}$, where $i \in [1, N]$ and $j \in [1, 3]$. Let Hash be a collision resistant hash function with output size 2λ bits. The CRS is $\mathsf{crs} = (\mathrm{ck}_{\mathbb{G}_{\mathsf{com}}}, \mathrm{ck}_{\mathbb{G}_{\mathsf{3sq}}})$.

4.2. Scheme Overview

The Σ-protocol $\mathsf{Sharp}_{\mathsf{GS}}$ is described in algorithm 1. The prover receives the witnesses $x_i \in [0, B]$ and $r_x \in [0, S]$, and the statement $C_x = r_x G_0 + \sum_{i=1}^{N} x_i G_i$ and B as input. Prover and verifier proceed as follows: (1) In the first flow, the prover computes and commits to a decomposition of x_i using MPed in $\mathbb{G}_{\mathsf{com}}$ (lines 1 and 2). Then, for all repetitions $k \in [1, R]$, she commits to random masks of the witnesses and decomposition in MPed over $\mathbb{G}_{\mathsf{com}}$ (line 4 to 7) and the garbage terms of the decomposition polynomial (lines 8 to 12). Finally, she sends the commitments to the verifier. (2) In the second flow, the verifier draws a random challenge for each repetition (line 1) and sends it to the prover. (3) In the third flow, the prover masks the witnesses (multiplied with the challenges) for each repetition and sends the result to the verifier (lines 13 to 18). (4) Finally, the verifier checks whether the linear relation between the commitments and the challenge holds, after recomputing the decomposition polynomial (lines 2 to 8).

Optimizations. We use uniform rejection sampling for the masking (instead of Gaussian rejection sampling in CKLR). This reduces the masking overhead in our setting. As in CKLR, the prover can avoid sending the commitments $\mathcal{D} = (D_{k,x}, D_{k,y}, D_{k,*})_{k=1}^{R}$ by replacing the output $\mathcal{D}$ in the first flow with a hash $\Delta \leftarrow \mathsf{Hash}(\mathcal{D})$. Then, the verifier can recompute $\mathcal{D}$ in the verification and check whether the hash matches. Applying the Fiat-Shamir transformation yields a non-interactive range proof.

4.3. Security and Correctness

Non-abort probability. With R repetitions, the probability of the honest prover *not* aborting (due to masking) is lower-bounded by $[(1 - \mathsf{p}_\mathsf{r})^3 \cdot (1 - \mathsf{p}_\mathsf{x})^{4N}]^R$.

Security. $\mathsf{Sharp}_{\mathsf{GS}}$ proofs satisfy correctness, non-abort SHVZK and relaxed soundness. Intuitively, the verifier is convinced that the committed value has a *unique* rational representative in the range $[-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}}$, formalized in theorem 4.1 below. Note that with the *four* square decomposition, we obtain exact range membership in $[0, B]$, in exchange for slightly increasing proof size (see section 6.1.2).

Theorem 4.1. *The scheme* $\mathsf{Sharp}_{\mathsf{GS}}$ *has correctness error at most* $1 - [(1 - \mathsf{p}_\mathsf{r})^3 \cdot (1 - \mathsf{p}_\mathsf{x})^N]^R$. *It is non-abort SHVZK under the* SEI *assumption in* $\mathbb{G}_{\mathsf{com}}$ *and* $\mathbb{G}_{\mathsf{3sq}}$. *If* $2(B\Gamma^2 + 1)L < p$ *and* $18K^2 < q$ *with* $K = (B\Gamma + 1)L$, *then* $\mathsf{Sharp}_{\mathsf{GS}}$ *has relaxed soundness under the* DLOG *and* SEI *assumptions in* $\mathbb{G}_{\mathsf{com}}$ *and* $\mathbb{G}_{\mathsf{3sq}}$ *with knowledge error* $(\frac{2}{\Gamma+1})^R$ *for the relation* $\mathsf{R}_{\mathsf{Ext}} = \{((x_i)_{i=1}^N, r_x) \colon C_x = r_x G_0 + \sum_{i=1}^{N} x_i G_i \wedge [x_i]_{\mathbb{Q}} \in [-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}_{K,\Gamma}}\}$. *To be precise, we consider*

Algorithm 1 $\mathsf{Sharp}_{\mathsf{GS}}$

Prover$(C_x, B, r_x, \{x_i\}_{i=1}^N)$ | Verifier(C_x, B)

Compute $y_{i,j}$ s.t. $4x_i(B-x_i)+1 = \sum_{j=1}^3 y_{i,j}^2$ for $i \in [1,N]$
Set $C_y = r_y G_0 + \sum_{i=1}^N \sum_{j=1}^3 y_{i,j} G_{i,j}$ for $r_y \xleftarrow{\$} [0,S]$
for all $k \in [1,R]$ **do**
  Set $\widetilde{r}_{k,x}, \widetilde{r}_{k,y} \xleftarrow{\$} \mathsf{R}_r$ ▷ Opening
  Set $\widetilde{x}_{k,i}, \widetilde{y}_{k,i,j} \xleftarrow{\$} \mathsf{R}_x$ for $i \in [1,N], j \in [1,3]$
  Set $D_{k,x} = \widetilde{r}_{k,x} G_0 + \sum_{i=1}^N \widetilde{x}_{k,i} G_i$
  Set $D_{k,y} = \widetilde{r}_{k,y} G_0 + \sum_{i=1}^N \sum_{j=1}^3 \widetilde{y}_{k,i,j} G_{i,j}$
  Set $r_k^* \xleftarrow{\$} [0,S]$ and $\widetilde{r}_k^* \xleftarrow{\$} \mathsf{R}_r$ ▷ Decomposition
  Set $\alpha_{1,k,i}^* = 4\widetilde{x}_{k,i} B - 8x_i \widetilde{x}_{k,i} - 2\sum_{j\in[1,3]} y_{i,j}\widetilde{y}_{k,i,j}$ for $i \in [1,N]$
  Set $\alpha_{0,k,i}^* = -(4\widetilde{x}_{k,i}^2 + \sum_{j\in[1,3]} \widetilde{y}_{k,i,j}^2)$ for $i \in [1,N]$
  Set $C_{k,*} = r_k^* H_0 + \sum_{i=1}^N \alpha_{1,k,i}^* H_i$
  Set $D_{k,*} = \widetilde{r}_k^* H_0 + \sum_{i=1}^N \alpha_{0,k,i}^* H_i$

$\xrightarrow{C_y, \{C_{k,*}, D_{k,x}, D_{k,y}, D_{k,*}\}_{k=1}^R}$

$\gamma_k \xleftarrow{\$} [0,\Gamma]$ for $k \in [1,R]$ ▷ Challenge

$\xleftarrow{\{\gamma_k\}_{k=1}^R}$

for all $k \in [1,R], i \in [1,N], j \in [1,3]$ **do**
  Set $z_{k,i} = \mathsf{mask}_x(\gamma_k \cdot x_i, \widetilde{x}_{k,i}), z_{k,i,j} = \mathsf{mask}_x(\gamma_k \cdot y_{i,j}, \widetilde{y}_{k,i,j})$
  Set $t_{k,x} = \mathsf{mask}_r(\gamma_k r_x, \widetilde{r}_{k,x}), t_{k,y} = \mathsf{mask}_r(\gamma_k \cdot r_y, \widetilde{r}_{k,y})$
  Set $t_k^* = \mathsf{mask}_r(\gamma_k \cdot r_k^*, \widetilde{r}_k^*)$
if any $z_{k,i}$, $t_{k,x}$ or t_k^* is $\perp$ **then**
  abort ▷ Masking failed

$\xrightarrow{\{z_{k,i,j}, z_{k,i}, t_{k,x}, t_{k,y}, t_k^*\}_{k\in[1,R], i\in[1,N], j\in[1,3]}}$

for all $k \in [1,R]$ **do**
  Check $D_{k,x} + \gamma_k C_x = t_{k,x} G_0 + \sum_{i=1}^N z_{k,i} G_i$
  Check $D_{k,y} + \gamma_k C_y = t_{k,y} G_0 + \sum_{i=1}^N \sum_{j=1}^3 z_{k,i,j} G_{i,j}$
  Set $f_{k,i}^* = 4z_{k,i}(\gamma_k B - z_{k,i}) + \gamma_k^2 - \sum_{j=1}^3 z_{k,i,j}^2$
  Check $D_{k,*} + \gamma_k C_{k,*} = t_k^* H_0 + \sum_{i=1}^N f_{k,i}^* H_i$
  Check $z_{k,i}, z_{k,i,j} \in [0, (B\Gamma+1)L_x]$ for $i \in [1,N], j \in [1,3]$
return 1 iff all checks succeed

the S-bounded SEI *assumption in* $\mathbb{G}_{\mathsf{com}}$ *and* $\mathbb{G}_{\mathsf{3sq}}$*. Moreover, in* $\mathsf{R}_{\mathsf{Ext}}$ *all* $[x_i]_{\mathbb{Q}}$ *have a common denominator* $d \in [1, \Gamma]$*.*

Security proof, outline. Here, we only sketch the proof of security and the relaxed soundness guarantee. We refer to appendix E.1 for details. (The proof is given for the $\mathsf{Sharp}_{\mathsf{GS}}$ with all optimizations.) Informally, the committed x_i are guaranteed to have rational representatives in $[-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}_{K,\Gamma}}$, where the numerator and denominator is bounded by $K = (B\Gamma + 1)L$ and Γ respectively.

Since either mask aborts or the z's lie within a predetermined range, correctness follows easily. Also, we can simulate a valid transcript of the proof for statement (C_x, B) by first sampling the challenge and then computing a transcript starting from the last flow. For this, we replace each witness w in the masking $\mathsf{mask}(\gamma w, \widetilde{w})$ with 0 (where $\widetilde{w}$ is the used mask) which affects the distribution only by $\varepsilon_{\mathsf{mask}} = 0$ (see section 2.5). If any masking aborts, the simulator returns $\perp$. Thus, the scheme is non-abort SHVZK under the SEI assumption (for hiding commitments). For the soundness proof, we show 3-special soundness, i.e. extraction from 3 related transcripts. First, we extract the commitments (with a standard argument). Second, we verify that the three square decomposition holds over $\mathbb{Z}_q$ for the extracted x_is and infer that $[x_i]_{\mathbb{Q}} \in [-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}}$ using lemma B.2. The switch between groups requires special care, as the rings $\mathbb{Z}_p$ and $\mathbb{Z}_q$ are "algebraically incompatible". But the shortness of the extracted values suffices to show that the three square decomposition over $\mathbb{Z}_q$ implies non-negativity for the rational representative committed over $\mathbb{Z}_p$.

5. $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$: Improved Proof of Short Opening

We present $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$, which is based on $\mathsf{Sharp}_{\mathsf{GS}}$ but uses a (batch) shortness test to separate PoSO and PoDec, and to reduce costs of "internal" repetitions.

5.1. Parameters

The groups $\mathbb{G}_{\mathsf{com}}$ and $\mathbb{G}_{\mathsf{3sq}}$, and parameters B, Γ, N, and S, are identical to $\mathsf{Sharp}_{\mathsf{GS}}$ (cf. section 4.1). The commitment key $\mathrm{ck}_{\mathsf{com}}$ is augmented by additional elements $\widetilde{G}_j \xleftarrow{\$} \mathbb{G}_{\mathsf{com}}$ for $j \in [1, R]$. For simplicity, we define $\widehat{\Gamma} \coloneqq (\Gamma + 1)^R - 1$ (the size of "large" challenge), and require that $\widehat{\Gamma} \leq p$.[11]

More concretely, we consider a *small group* $\mathbb{G}_{\mathsf{com}}$ of order p and a *large group* $\mathbb{G}_{\mathsf{3sq}}$ (which may be equal) of order q. Let B be a *range bound* and let Γ be a bound for the *challenge sizes*. Let N be the *batch size*, i.e. the number of committed x_i whose range membership is to be proven. Let R be the number of "internal repetitions" of the Batch-PoSO. Let $\widehat{\Gamma} \coloneqq (\Gamma + 1)^R - 1$ be the size of "large" challenges, and assume that $\widehat{\Gamma} \leq p$.

Commitment key setup. The commitment key is $\mathrm{ck} = (\mathrm{ck}_{\mathsf{com}}, \mathrm{ck}_{\mathsf{3sq}})$, where

- $\mathrm{ck}_{\mathsf{com}} = (\{G_i\}_{i\in[0,N]}, \{G_{i,j}\}_{i\in[1,N],j\in[1,3]}, \{\widetilde{G}_j\}_{j\in[1,R]})$, where $G_i, G_{i,j}, \widetilde{G}_j \xleftarrow{\$} \mathbb{G}_{\mathsf{com}}$.
- $\mathrm{ck}_{\mathsf{3sq}} = (\{H_i\}_{i\in[0,N]})$, where $H_i \xleftarrow{\$} \mathbb{G}_{\mathsf{3sq}}$.

The elements G_0 and H_0 are used for random masking terms of the commitment. The elements $G_1, \ldots, G_N$ are used to commit to x_i, $G_{i,1}, \ldots, G_{i,3}$ are used for the 3-square decomposition $y_{i,1}, \ldots, y_{i,3}$ of $1 + 4x_i(B - x_i)$, and $\widetilde{G}_1, \ldots, \widetilde{G}_R$ are for Batch-PoSO masks μ_j. The elements $H_1, \ldots, H_N$ are used to commit to the garbage terms for linearization of the square decomposition proof.

[11] Since the maximal challenge set for a scalar challenge is $[0, p-1] = \mathbb{Z}_p$, increasing the challenge set would require repetitions in "Phase 2", which is trivially implemented but completely unnecessary for our instantiations.

Masking and mask sizes. For simplicity, we fix a single masking overhead L for all masks. Logically, some masks must be short due to shortness checks, while other masks only hide the value and shortness is used to reduce communication. The latter may be drawn uniformly from $\mathbb{Z}_p$ as well. In $\mathsf{Sharp}_{\mathsf{GS}}$, L_x was the former, L_r the latter type. In $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$, we have following masking behaviour:

- $\mathsf{R}_{poso} = [0, (V_{poso} + 1)L]$, where $V_{poso} = 4NB\Gamma$ must be short.
- For $z \in \{x, \mu, r, r^*\}$, R_z need only hide the value, so $\mathsf{mask}_z(v, m)$ is *computed modulo* p (resp. q). If $\mathsf{R}_z = \mathbb{Z}_p$ (resp. $\mathbb{Z}_q$), mask_z never aborts.
- For $z \in \{x, \mu, r\}$, we set $\mathsf{R}_z = [0, \min(p-1, (V_z+1)L)]$, where $V_x = B\widehat{\Gamma}$, $V_r = S$, and $V_\mu = \mathsf{R}_{poso} \cdot \widehat{\Gamma}L$that is, $V_\mu = (V_{poso}+1)L \cdot \widehat{\Gamma}L$. And we set $\mathsf{R}_{r^*} = [0, \min(q-1, (V_{r^*}+1)L)]$ where $V_{r^*} = S$.
- If $\mathbb{G}_{\mathsf{com}} = \mathbb{G}_{\mathsf{3sq}}$, then typically $\mathsf{R}_r = \mathsf{R}_{r^*} = \mathsf{R}_\mu = \mathbb{Z}_p$.

5.2. Scheme Overview

The difference between $\mathsf{Sharp}_{\mathsf{GS}}$ and $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ is the use of the Batch-PoSO. Again, to simplify we only consider one range $[0, B]$ for all x_i. It will be evident how to generalize to independent ranges $x_i \in [0, B_i]$.

The scheme is defined in algorithms 2 and 3. It is a 5-move protocol which effectively consists of 2 phases: In Phase 1, the prover commits to the 3-square decompositions (and masks μ_k). Then, k parallel random affine shortness tests are run on committed values. In Phase 2, the prover proves that it has correctly answered the shortness test, and that the 3-square decomposition holds modulo q. Thus, Phase 2 is very similar to $\mathsf{Sharp}_{\mathsf{GS}}$, except, it uses a large challenge space $[0, \widehat{\Gamma}]$, so no repetitions are required.

5.3. Security and Correctness

Non-abort probability. With R "internal" repetitions, the number of masking operations are R in Phase 1. In Phase 2, we have $4N$ for x_i and $y_{i,j}$, again R for μ_k, and 3 masks for r_x, r_y, r^*. Thus, the probability of the honest prover *not* aborting (due to masking) is lower-bounded by $(1 - \frac{1}{L})^{2R+4N+3}$.

Security. The security guarantee of $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ is almost the same as that of $\mathsf{Sharp}_{\mathsf{GS}}$, except for a small tightness loss due to the weaker (provable) guarantees of the shortness test (theorem 3.3).

Theorem 5.1. *The scheme* $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ *has correctness error at most* $1 - (1 - \frac{1}{L})^{2R+4N+3}$. *It is non-abort SHVZK under the* SEI *assumption in* $\mathbb{G}_{\mathsf{com}}$ *and* $\mathbb{G}_{\mathsf{3sq}}$. *Let* $K' = (1 + 2\beta)K$ *where* $K = (B\Gamma + 1)L$ *and* $\beta = \min(4N, \mathsf{primlmin}(\Gamma + 1))$. *If* $18(K')^2 < q$ *and* $2(\Gamma+1)^2K' < p$ *and* $(\Gamma+1)^R - 1 < p$, *then* $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ *has relaxed soundness under the* DLOG *and* SEI *assumptions in* $\mathbb{G}_{\mathsf{com}}$ *and* $\mathbb{G}_{\mathsf{3sq}}$ *with knowledge error* $\frac{2+8^R}{(\Gamma+1)^R}$ *for the relation* $\mathsf{R}_{\mathsf{Ext}} = \{((x_i)_{i=1}^N, r_x) \colon C_x = r_x G_0 + \sum_{i=1}^N x_i G_i \wedge [x_i]_{\mathbb{Q}} \in [-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}_{K',\Gamma}}\}$. *To be precise, we consider the* S*-bounded* SEI *assumption in* $\mathbb{G}_{\mathsf{com}}$ *and* $\mathbb{G}_{\mathsf{3sq}}$. *Moreover, in* $\mathsf{R}_{\mathsf{Ext}}$ *all* $[x_i]_{\mathbb{Q}}$ *have a common denominator* $d \in [1, \Gamma]$.

Security proof, outline. The proof of correctness and non-abort SHVZK for $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ are completely analogous to the respective proofs for $\mathsf{Sharp}_{\mathsf{GS}}$.

The ideas behind the soundness proof of theorem 5.1 are quite straightforward. It proceeds by dealing with the two phases separately. First, observe that Phase 2 is effectively a Σ-protocol

Algorithm 2 $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$– Phase 1

Prover$(C_x, B, r_x, \{x_i\}_{i=1}^N)$		Verifier(C_x, B)
1: Compute $4x_i(B-x_i)+1 = \sum_{j=1}^3 y_{i,j}^2$ for $i \in [1,N]$		
2: Set $r_y \xleftarrow{\$} [0,S]$ and $\mu_1, \ldots, \mu_R \xleftarrow{\$} \mathsf{R}_{poso}$		
3: Set $C_y = r_y G_0 + \sum_{i=1}^N \sum_{j=1}^3 y_{i,j} G_{i,j} + \sum_{k=1}^R \mu_k \widetilde{G}_k$		
	$\xrightarrow{C_y}$	
		1: Sample $\gamma_{i,j}^{(k)} \xleftarrow{\$} [0,\Gamma]$ for $i \in [1,N]$, $j \in [0,3]$, $k \in [1,R]$
	$\xleftarrow{\{\gamma_{i,j}^{(k)}\}_{i,j,k}}$	
4: Let $y_{i,0} \coloneqq x_i$		
5: Set $\zeta_k \coloneqq \mathsf{mask}_{poso}(\sum_{i=1}^N \sum_{j=0}^3 \gamma_{i,j}^{(k)} y_{i,j}, \mu_k)$ for $k \in [1,R]$		
6: **if** any ζ_k is $\perp$ **then**		
7: **abort** ▷ Masking Failed		
	$\xrightarrow{\{\zeta_k\}_{k\in[1,R]}}$	
		2: **if** any $\zeta_k \notin [0,(4NB\Gamma+1)L]$ **then**
		3: **return** 0 ▷ PoSO rejected

Run Phase 2: Proof of consistency of ζ_k and 3-square decomposition (see algorithm 3)

for the statement which was completed in Phase 1, i.e. that C_x resp. C_y are commitments to the x_i's resp. auxiliary values $y_{i,j}$ and μ_k, the answers ζ_k of a random affine shortness test are correct, and the 3-square decomposition holds. Indeed, Phase 2 is 3-special sound, i.e. given 3 accepting transcripts identical up until the challenge message γ for 3 distinct challenges, one can extract openings to the commitments which satisfy the relation (or the binding property is broken). Thus, as a first step, one can replace Phase 2 with an extractor with knowledge error $2/(\Gamma+1)^R$.

Next, one needs to argue that the x_i and $y_{i,j}$ are short (from above, we know that they satisfy the 3-square decomposition). This does not follow from (3 transcripts for) Phase 2 alone. Intuitively, if the "shortness test" used has soundness error κ, then if any x_i, $y_{i,j}$ is not short, the probability that the verifier accepts is at most κ^R. More precisely, if there is no $d \in [1,\Gamma]$ such that $dx_i, dy_{i,j} \in [-K', K']_{\mathbb{Z}_p}$ for all i,j, then the shortness test accepts with probability at most κ. However, there is a gap: Our commitment is only computationally binding, so, by breaking the commitment, the adversary might win with probability ε (much) higher than κ^R. Fortunately, to win with probability $\varepsilon > \kappa^R$, the adversary *must* break the binding property. Thus, except with probability κ^R, one obtains a binding break from such an adversary in expected time (by rewinding until $\mathscr{A}$ succeeds again). Overall, this proves the soundness claim of theorem 5.1.

5.4. Trade-offs and Optimizations

Reducing communication. As with $\mathsf{Sharp}_{\mathsf{GS}}$, hashing can reduce the communication in Phase 2 of the protocol. Namely, the final verification step in Phase 2 computes F_x, F_*, $\{f_k\}_{k\in[1,R]}$, and checks if they are equal to D_x, D_*, $\{d_k\}_{k\in[1,R]}$. This check can be compressed by using a

Algorithm 3 $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$– Phase 2

After Phase 1 (shortness proof, see algorithm 2)

Set $\widetilde{r}_x, \widetilde{r}_y \xleftarrow{\$} \mathsf{R}_r$
Set $\widetilde{x}_i, \widetilde{y}_{i,j} \xleftarrow{\$} \mathsf{R}_x$ for $i \in [1, N], j \in [1, 3]$
Set $\widetilde{\mu}_k \xleftarrow{\$} \mathsf{R}_\mu$ for $k \in [1, R]$ ▷ PoSO
Set $d_k = \sum_{i=1}^{N} \sum_{j=0}^{3} \widetilde{y}_{i,j} \gamma_{i,j}^{(k)} + \widetilde{\mu}_k$ for $k = 1, \ldots, R$ ▷ PoSO
Set $D_x = \widetilde{r}_x G_0 + \sum_{i=1}^{N} \widetilde{x}_i G_i$
Set $D_y = \widetilde{r}_y G_0 + \sum_{i=1}^{N} \sum_{j=1}^{3} \widetilde{y}_{i,j} G_{i,j} + \sum_{k=1}^{R} \widetilde{\mu}_k \widetilde{G}_k$
Set $r^* \xleftarrow{\$} [0, S]$ and $\widetilde{r}^* \xleftarrow{\$} \mathsf{R}_{r^*}$
Set $\alpha^*_{1,i} = 4\widetilde{x}_i B - 8 x_i \widetilde{x}_i - 2 \sum_{j \in [1,3]} y_{i,j} \widetilde{y}_{i,j}$ for $i \in [1, N]$
Set $\alpha^*_{0,i} = -(4\widetilde{x}_i^2 + \sum_{j \in [1,3]} \widetilde{y}_{i,j}^2)$ for $i \in [1, N]$
Set $C_* = r^* H_0 + \sum_{i=1}^{N} \alpha^*_{1,i} H_i$
Set $D_* = \widetilde{r}^* H_0 + \sum_{i=1}^{N} \alpha^*_{0,i} H_i$

$\xrightarrow{C_*, D_x, D_y, D_*, \{d_k\}_{k=1}^{R}}$

Set $\gamma \xleftarrow{\$} [0, (\Gamma + 1)^R - 1) \subseteq \mathbb{Z}_p$ ▷ Large challenge

$\xleftarrow{\gamma}$

for all $i \in [1, N]$, $j \in [1, 3]$, $k \in [1, R]$ **do**
    Set $z_i = \mathsf{mask}_x(\gamma \cdot x_i, \widetilde{x}_i)$ and $z_{i,j} = \mathsf{mask}_x(\gamma \cdot y_{i,j}, \widetilde{y}_{i,j})$
    Set $t_x = \mathsf{mask}_r(\gamma \cdot r_x, \widetilde{r}_x)$ and $t_y = \mathsf{mask}_r(\gamma \cdot r_y, \widetilde{r}_y)$
    Set $t^* = \mathsf{mask}_r(\gamma \cdot r^*, \widetilde{r}^*)$
    Set $\tau_k = \mathsf{mask}_\mu(\gamma \cdot \mu_k, \widetilde{\mu}_k)$ ▷ PoSO
if any z_i, $z_{i,j}$, t_x, t_y t^*, τ_k is $\perp$ **then**
    abort ▷ Masking failed

$\xrightarrow{\{z_i\}_{i \in [1,N]}, \{z_{i,j}\}_{i \in [1,N], j \in [1,3]}, t_x, t_y, t^*, \{\tau_k\}_{k \in [1,R]}}$

Compute $F_x = -\gamma C_x + t_x G_0 + \sum_{i=1}^{N} z_i G_i$
Compute $F_y = -\gamma C_y + t_y G_0 + \sum_{i=1}^{N} \sum_{j=1}^{3} z_{i,j} G_{i,j} + \sum_{k=1}^{R} \tau_k \widetilde{G}_k$
Let $z_{i,0} \coloneqq z_i$
Set $f_k = -\gamma \zeta_k + \sum_{i=1}^{N} \sum_{j=0}^{3} z_{i,j} \gamma_{i,j}^{(k)} + \tau_k$ for $k \in [1, R]$ ▷ PoSO
Compute $f_i^* = 4 z_i (\gamma B - z_i) + \gamma^2 - \sum_{j=1}^{3} z_{i,j}^2$ for $i \in [1, N]$
Recompute $F_* = -\gamma C_* + t^* H_0 + \sum_{i=1}^{N} f_i^* H_i$
if $F_x = D_x$, $F_y = D_y$, $F_* = D_*$, and $f_k = d_k$ for $k \in [1, R]$ **then**
    return 1
else return 0

collision resistant hash function H, and having the prover send $D_H = H(D_x, D_y, D_*, \{d_k\}_{k\in[1,R]})$ instead. The verification now checks $D_H = F_H$, where $F_H \coloneqq H(F_x, F_*, \{f_k\}_{k\in[1,R]})$. It is easy to see that the protocol remains secure if H is collision resistant. Also, since Phase 2 is effectively independent of Phase 1, it may be exchanged with other suitable (succinct) argument systems. This is discussed in a later paragraph.

Fiat–Shamir transformation. $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ is public-coin and the Fiat–Shamir transformation is applicable. This yields a *non-interactive* zero-knowledge argument. If the (hash) function is modelled as a random oracle, the resulting scheme is provably secure in the ROM, although there is a security loss (in the number of random oracle queries).

As $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ is not a usual Σ-protocol, nor special sound (with sensible parameters), well-known extraction techniques are not directly applicable. However, the reasoning for the security of the Fiat–Shamir transformation of multi-round special sound protocols in recent works [AFK21; Wik21] should be applicable to our setting. After all, the step in Phase 1 is not particularly involved, and we have a property akin to special soundness there: If a second transcript is necessary (due to inconsistent witness extracted in Phase 2), then a uniformly *random* accepting transcript (with same message) will, with high probability, lead to a non-trivial DLOG relation.

Proving non-negativity. As with CKLR proofs [Cou+21a], it is possible to only prove $x \geq 0$ instead of $x \in [0, B]$. Namely, using $1 + 4x = \sum_{i=1}^{3} y_i^2$ shows $x \geq -1/4$, and using the four square decomposition shows $x \geq 0$. This is of interest if the upper bound B is "unreachable" or otherwise not of interest. However, an upper bound B for x is still required (and must not be too large), as it determines the size of the masks and the verifier's size checks as before (since wrap-around must still be prevented). Moreover, B is the maximal value for which zero-knowledge guarantees hold; the larger $x > B$ becomes, the more zero-knowledge degrades. This optimization applies to $\mathsf{Sharp}_{\mathsf{GS}}$ and $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$.

Standard Soundness and higher knowledge error. It is easy to see that RAST with uniform distribution over $\{0,1\}^N$ is fractionally $(NBL, 1)$-sound with error $1/2$. In this case, $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ has standard soundness with knowledge error $\kappa_{\mathsf{err}} = 2^{-R}$, and R repetitions require approximately $2R \cdot \log(NBL)$ bits communication overhead. This trade-off is especially interesting if high knowledge error is acceptable. for example, a statistical knowledge error $\kappa_{\mathsf{err}} = 2^{-40} + \mathrm{negl}$ in interactive settings[12] is a common choice, and in application to anonymous credentials may be considered acceptable.

By using the Fiat–Shamir transformation on Phase 2 (with $\widehat{\Gamma} = 2^{\lambda} - 1$), an interactive 3-move protocol can be obtained.[13] The trade-off is also useful if batch size N is huge (hence amortized cost to achieve standard soundness is small). In that case, exchanging Phase 2 is also of interest.

Exchanging phase 2. Since Phase 2 is effectively independent of Phase 1, i.e. the shortness test, it may be exchanged with other suitable (succinct) argument systems. This is especially interesting to reduce overall communication. As only knowledge of openings and simple quadratic equations are proven, generic proof systems which target R1CS over $\mathbb{Z}_p$ (e.g. Bulletproofs [Boo+16; Bün+18]) or general quadratic or polynomial equations over $\mathbb{Z}_p$ (e.g. [HKR19; BG18]) can be used as drop-ins.

In fact, the ζ_k could also be committed to and proven to be computed correctly and that they lie within $[0, K]$; if done in zero-knowledge, this makes the masking terms μ_k superfluous,

[12] In this case, the communication overhead is reasonable and computational efficiency remains excellent. For 128 repetitions, the communication overhead becomes noticeable. See table 4 for concrete size estimates.

[13] We stress that high knowledge error, e.g. 2^{-40}, only makes sense in interactive settings. Fiat–Shamir transformations are trivial (and cheap) to break in this regime.

improving the soundness error of the shortness test. However, a (standard sound) range proof is needed to check $\zeta_k \in [0, 4NB\Gamma]$, and the proof system must now include adaptively chosen commitments and statements, which typically is not a problem for commit-and-prove-based proof systems, but it does slightly increase proof size and round complexity. Considering the number of variables for adding the necessary constraints for the (bit-decomposition) range proof for ζ_k, and using binary challenges in the Batch-PoSO (to obtain standard soundness) with $R = 128$ repetitions for security, we obtain a break-even in the number of (auxiliary) variables at about $N = 170$ with naive R1CS-type constraints (and $N = 160$ for quadratic constraints). Overall, for large enough batches sizes, this approach may be of interest. See example 5.2 for more details.

Example 5.2 (Using a succinct Phase 2). As discussed in section 5.4, it is possible to adapt suitable succinct arguments which follow a commit-then-prove strategy which allow multiple commitment steps and an adaptive choice of the final statement. The upside is, that the 3-square decomposition requires fewer auxiliary variables (compared to bit-decomposition). The downside is, that an overhead which is almost independent of the batch size must be paid (namely, the R repetitions). We discuss and roughly quantify this trade-off, where we use the PoSO with binary challenges to achieve standard soundness. For concreteness, consider the Bulletproofs variant [HKR19], which allows proving quadratic equations over committed variables (instead of the weaker R1CS-type equations). We set $B = 2^{64} - 1$ and $R = 128$ (and $\Gamma = 1$).

The protocol with exchanged Phase 2 works as follows:

1. (Phase 1) Commit not only to x_i, but also to the square decomposition $y_{i,j}$ and masks μ_k for $k = 1, \ldots, R$.
2. Receive the PoSO challenges and responds with ζ_k.
3. (Phase 2) Both prover and verifier adapt the statement by including the linear check constraints (for $k = 1, \ldots, R$) for the PoSO, similar to Phase 2 of $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$.

With this approach, sending $\{\zeta_k\}_{k=1}^{R}$ significantly increases the proof size (namely, by R elements of $\log((4N\Gamma B + 1)L)$ bits each). But only 4 variables per range are used (x_i and 3 auxiliary variables $\{y_{i,j}\}_j$), whereas 64 variables are required per bit-decomposition.

A more complex approach enables smaller proofs, but requires an adaptive commitment and statement:

1. Commit not only to x_i, but also to the square decomposition $y_{i,j}$.
2. Receive the PoSO challenges and *commit* to ζ_k for $k \in [1, R]$.
3. Both prover and verifier adapt the statement by including the linear check constraint (for $k = 1, \ldots, R$) for the PoSO, and a (bit-decompostion) range proof for $\zeta_k \in [0, 4N\Gamma B]$.

An advantage is, that no masks are necessary, but we now require an adaptive commitment to ζ_k which still slightly increases proof size. For the normal bit-decomposition of all x_i, one needs $N \log(B + 1)$ variables and quadratic constraints. With this approach, one needs $4N + R \cdot \log(4N\Gamma B + 1)$ variables and quadratic constraint (plus a suitable commit-and-prove system). For $R = \lambda = 128$ and $B = 2^{64} - 1$ the break-even point in terms of variables and constraints is at about $N = 160$, and at $N = 2048$ we observe an over 7-fold reduction in terms of variables and constraints, which tends to 16 as N grows. For R1CS-type constraints (now using 8 instead of 4 variables), we observe break-even at about $N = 170$ and an almost 5-fold reduction at $N = 2048$ which tends to 8 as N grows.

6. Soundness Guarantees and Hidden Order Augmentation

We provide some insights into the consequences of relaxed soundness and the use of hidden order groups in that context. Further discussions can be found in appendix B and appendix C respectively.

6.1. Remarks on Relaxed Soundness

The relaxed soundness of CKLR-type proofs only ensures that a committed value x is a fraction $x \equiv_p m/d$ with short numerator and denominator, say $x \in \mathbb{Q}_{M,D}$. As we will see, this can be sufficient in important applications, such as anonymous credentials. However, this guarantee is, in general, too weak to allow unchecked homomorphic operations on commitments, e.g. the sum $\sum_{i=1}^{N} \frac{m_i}{d_i}$ of short fractions m_i/d_i need not be short. The main problem is the growth of the common denominator as $d = \mathrm{lcm}(d_1, \ldots, d_N)$, and the numerator grows similarly. Thus, after a few operations, all guarantees on shortness are lost.

6.1.1. Cheating with Small Denominators.

The use of relaxed soundness is *not* a proof artefact: For small d and m, find $\sum_{j=1}^{3} a_j^2 = d^2 + 4(m-d)m$ and let $x \equiv_p m/d$ and $y_j \equiv_p a_j/d$. This decomposition has a chance of $1/d$ (per repetition, and $1/d^R$ overall) to fool the verifier. In particular, after the Fiat–Shamir transformation, generating proofs for x is efficiently possible if d is not too large.

6.1.2. Three Square Decomposition.

Our range proofs use the 3-square decomposition and prove membership in $[-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}_{K',\Gamma}}$. To obtain membership in $[0, B]_{\mathbb{Q}_{K',\Gamma}}$ one can either use the 4-square decomposition, or use $\Gamma < 4B$ (perhaps, increasing repetitions), as this ensures that denominators $d \geq 4B$ violate soundness, hence $[0, B]_{\mathbb{Q}_{K',\Gamma}} = [-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}} \cap \mathbb{Q}_{K',\Gamma} = [-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}_{K',\Gamma}}$.

6.2. Using Groups of Hidden Order

The problem of denominator growth can be mitigated by resorting to a group $\mathbb{H}$ of hidden order. For $\mathsf{Sharp}_{\mathsf{GS}}$ and $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$, the approach works as follows: Add a single additional commitment C'_x to all values x_i in $\mathbb{H}$ (using a MPed commitment). Moreover, include a proof of knowledge of opening of C'_x (to the same value as in C_x). This small change, allows us to reduce to properties of $\mathbb{H}$ to control the denominator. Using reasonable assumptions, it can be shown that the denominators d_i are of the form $d_i = e^{k_i}$ for $k_i \in \mathbb{N}_0$.

6.2.1. Instantiating the Hidden Order Group.

When instantiating $\mathbb{H}$ with suitable class groups of hidden order for which a plausible strengthened 2-fROOT assumption holds, the prover will be bound to dyadic rationals, i.e. x_i of the form $x_i = m_i/2^{k_i}$. This improves the applicability of the range proof significantly, since, even in homomorphic computations, the common denominator d is of the form 2^k with $k \leq \log(\Gamma)$. This restriction already enables the use of homomorphic computations.

When using RSA groups (with trusted setup), the proof provides standard soundness, since the prover is bound to an integer under the 1-fROOT assumption (a.k.a. strong RSA assumption). Interestingly, even without trusted setup, e.g. in cases with a "designated verifier", we sketch how RSA groups enable the use of Sharp proofs (cf. section 7.3).

We refer to appendix C for a more detailed overview of these "augmented" schemes with an efficiency and security analysis.

6.3. Non-Relaxed Soundness from Prior Knowledge

Prior knowledge on the shortness of committed values can "upgrade" the soundness from relaxed to non-relaxed. Namely, suppose for some reason, that you have prior knowledge or the guarantee that the committed value $x \in \mathbb{Z}_p$ is short, i.e. $x \in [-M, M]$. Then its representative in $\mathbb{Q}_{M,D}$ is an integer (namely, $\frac{x}{1}$). Thus, the range proof then directly implies that $x = [x]_{\mathbb{Q}} \in \mathbb{Z}$ is in the desired range $[0, B]_{\mathbb{Q}}$. More formally, we use that $[-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}} \cap \mathbb{Q}_{M,D} \cap \mathbb{Z} = [0, B]_{\mathbb{Q}} \cap \mathbb{Z} = [0, B]_{\mathbb{Z}}$. Note that this reasoning also works for the range proofs from CKLR [Cou+21a].

7. Applications

In this section, we show how range proofs with relaxed soundness, such as Sharp (or CKLR), can be used in certain applications, namely as anonymous credentials and anonymous transactions.

7.1. Anonymous Credentials

Anonymous credential schemes [Cha90; CL01; Bra00] allow users to obtain credentials from issuing authorities. Later, the user can present this credential to a verifier, without revealing his identity, which is fixed (but hidden via a commitment) in the credential. These credentials can also have attributes, for example a birthdate or a validity date. When showing the credential, the user might need to show that he is older than 18 or that the credential is still valid in a privacy-preserving manner.

Constructions of anonymous credentials typically rely on very efficient special-purpose zero-knowledge proofs. Concretely, most rely on so-called "CL-type" (algebraic) signature schemes, which come with very efficient proofs of knowledge of a signature on committed messages [CL03]. These are used to sign the identity and attributes of a user. To prove that attributes lie in some range, e.g. for age restrictions or a validity date of the credential, range proofs are employed. Thus, range proofs often constitute a significant, if not dominant, part in computation (and communication) in these settings.

Sharp proofs can often be used as an almost drop-in replacement in such settings. Consider the DLOG setting in a group of prime order p.

- When *issuing* the credential, all attribute values are known to the issuer. Assuming suitably small ranges $[0, B] \subseteq [-K, K]$ for valid attributes, the verifier's validity check of attribute values ensures shortness. If $K < p/(4\Gamma)$, then a rational representative m/d of an attribute x must be of the form $m/1$, i.e. x is a short integer. Thus, our range proof will be standard sound for x (see section 6.3).

- In case of *blind issuance* (where identity and attributes remain (partially) hidden), the relaxed soundness of DLOG-based Sharp *may not* suffice (see section 6.1.1). Here, we can use $\mathsf{Sharp}_{\mathsf{RSA}}$ which provides standard soundness, using a trusted public RSA-based setup of the issuer.

- For *showing* the credential, our range proofs can be used if the (blind) issuing phase ensured that the attributes lie within valid ranges, as in that case, our range proof is standard sound (see section 6.3).

The same reasoning applies to so-called *keyed-verification* anonymous credentials [CMZ14], where the issuer and verifiers have a shared secret key, which allows for more efficient protocols (but restricts the use-cases).

Anonymous credentials and their constructions come in many flavours [RVH17; CR19; BL13], and not all rely on prime order groups alone. Some use pairing groups and some use hidden order

groups. Nevertheless, it is very likely that in all these settings, our range proofs offer favourable trade-offs when compared to those in use. For example, while hidden order groups allow for three-square decomposition based range proofs, working in prime order groups is typically more efficient in terms of computation and communication. In the pairing-based settings, the approach of [CCs08] allows quite efficient digit-based decompositions. However, operations in pairing-groups are slower, elements are bigger, and for efficiency, [CCs08] needs relatively large (non-transparent) public parameters.

7.2. Updatable Anonymous Credentials and BBAs

A line of works [JR16; Har+17; Blö+19; Hof+20; Bob+20] uses techniques from anonymous credentials in a "non-static" manner to construct *updateable anoymous credentials* or *black-box accumulation (BBA) schemes*, which can be used for electronic payments, ticket systems, incentive systems and more. Most of the schemes feature range proofs as a core component, as these are required to prevent users from spending more than they have. The *(blind) issuing* process is mostly unchanged in comparison to anonymous credentials. The *show* protocol is replaced by (one or more) *update* protocol(s), which modify the user's attributes (e.g. the user's current balance).

Most applications work in the "public balance update" setting, where the user interacts with an operator, and the operator knows the amount Δ by which a user's (hidden) balance v is changed. That is, after the transaction, the balance should be $v + \Delta$, and for security, $v + \Delta \geq 0$ must be ensured. In this "public balance update" setting, our range proofs are again almost drop-in replacements. Namely, if the security proof ensures that the balance v is "small" (i.e. has rational representative $v/1$), then our proof has standard soundness for $v + \Delta \in [0, B]$. Since the security proofs typically prove inductively that, after each operation, the (new) balance v has certain properties (e.g. lies in the range $[0, B]$), the requirement for our proof to be standard sound is easily seen to be satisfied.

Range proofs are so expensive that early works [JR16; Har+17] consider weakened (security) requirements to achieve practical efficiency. Even in later works [Blö+19; Hof+20; Bob+20], they amount to a large part of (or even dominate) the runtime. Our optimized range proofs greatly improve efficiency.

7.3. Anonymous Transactions

Range proofs are often used in privacy-preserving blockchain-based smart contract platforms in order to ensure that the fixed (but hidden) balance of users is non-negative after performing a transfer [Zca; Mon; Bün+20]. This ensures that no user can spend more coins than he owns while preserving privacy. Thus, this is a "secret balance update" setting. We use the framework Zether [Bün+20] as a running example. In the following, we first give a short (and simplified) overview of Zether. Then, we showcase how problems of relaxed soundness from appendix B surface here and show where and how we can still apply Sharp in order to improve efficiency and communication. Lastly, we argue that augmented Sharp proofs (appendix C) suffice as drop-in replacement, either via a RSA commitment with trusted setup or *both* a RSA commitment and class group commitment without trusted setup. Interestingly, we can leverage the properties of RSA groups, even without trusted setup.

Overview of Zether. Let $B = 2^{32} - 1$. We refer to the cryptocurrency ZTH as coins which can be stored and transferred by users of the system (identified by a public key) [ElG84; CGS97]. The balance of the user is encrypted using (additively homomorphic) exponential ElGamal encryption. An encryption of balance $b \in [0, B]$ has the form $(C, R) = (bG + rY, rG)$, where $Y = xG$ is the public key, $x \in \mathbb{Z}_p^\times$ is the private key, $r \in \mathbb{Z}_p^\times$ is some randomness and $G \in \mathbb{G}$

for some group $\mathbb{G}$ with prime order p. For decryption, the user has to brute-force the discrete logarithm of bG to retrieve b.

After setup, a user has access to some methods in order to interact with a smart contract (that maintains the state st). User methods of interest are:

- $\mathsf{ReadBalance}(x, \mathsf{st})$: Given secret key x and state st, return the current balance (by decrypting the ElGamal ciphertext).
- $\mathsf{CreateFundTx}(Y, a)$: Given public key Y and amount a, creates a transaction tx that funds the user corresponding to Y with a coins. When the smart contract handles the transaction tx, it adds $a \geq 0$ to the balance (C, R) of user Y by setting $C \leftarrow C + aG$. Note that in the transaction, the amount a is public and thus no range proof is needed. The smart contract ensures that at most B coins are in the system when funding accounts, so overflows when funding are not possible.
- $\mathsf{CreateTransferTx}(x, Y', a, \mathsf{st})$: Takes as input secret key x of user with public key Y with balance (C, R), a public key Y' of a user with balance (C', R'), an amount a and the state st. Creates a transaction tx that transfers a coins from user Y to user Y' in a privacy-preserving manner.

  The transaction does not reveal the transferred amount a, since it contains encryptions of the balance a in (D, S) and (D', S') under public key Y and Y' respectively. The transaction further contains a zero-knowledge proof that both ciphertexts (D, S) and (D', S') encrypt the same value a, knowledge of the balance b in ciphertext (C, R), and two range proofs showing that $b - a$ and a are in the range $[0, B]$ (using Bulletproofs [Bün+18]). This ensures that the user cannot spend more coins than he owns.
- $\mathsf{CreateBurnTx}(x, \mathsf{st})$: Takes the state st as input and proves that the balance of user with secret key x is equal to amount a and removes a coins from the balance. This method withdraws all coins from an account.

There are also user methods to create an user address, and to lock and unlock an account. We focus on the methods listed above as they allow managing the balance of a user (for which range proofs are necessary). Further, there are several additional security mechanisms. For example signatures on transactions to prevent replay attacks, account locks and a pending transfer mechanism that ensures consistency. We refer to [Bün+20] for more details.

Problems with a Naïve Application of $\mathsf{Sharp}_{\mathsf{GS}}$. Additions of rational representatives $\mathbb{Q}_{N,D}$ without overflows can only be guaranteed if the number of additions is restricted (see appendix B). Further, correctness is only guaranteed for integers $\mathbb{Q}_{N,D} \cap \mathbb{Z}$. Thus, we cannot replace all range proofs in Zether with $\mathsf{Sharp}_{\mathsf{GS}}$. We elaborate:

- *Money creation I:* Using the three square decomposition, the committed rational representative is guaranteed to be in $[-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}}$ (see appendix B.2). So transferring $-1/(4B)$ is not (provably) prevented, though negative transfers are forbidden.
- *Money creation II:* With the four square decomposition, performing a range proof for a negative value has negligible probability of success. A user with balance b could still potentially create coins by creating n other accounts and transferring himself the amounts $a_i = m_i/p_i$ from account $i \in [1, n]$, where $\{p_i\}_{i=1}^n$ are distinct primes (see also section 6.1.1). After these transactions were applied, the users balance will be equal to $[b']_{\mathbb{Q}}$ with $b' \equiv_p b + \sum_{i=1}^n m_i/p_i$. If $\prod p_i > \Gamma$, the denominator overflows and the resulting balance is inconsistent (see appendix B.1). Indeed, $\sum_{i=1}^n m_i/p_i$ may be huge but within bounds.

- *Denial of service:* A malicious user can transfer an amount $a = \frac{1}{p}$ to a user with balance b. After the transaction, the receiving user's balance is $b' = \frac{pb+1}{p} \notin [0, B]_{\mathbb{Z}} \subseteq \mathbb{Z}$. This breaks correctness for honest users.

Applications of Sharp in Zether. Despite the problems sketched above, we can use Sharp proofs in Zether. For $\mathsf{Sharp}_{\mathsf{GS}}$, we ensure a shortness guarantee for the balances of users by combining it with homomorphic range proofs with standard soundness guarantees (such as Bulletproofs). In this case, $\mathsf{Sharp}_{\mathsf{GS}}$ satisfies standard soundness. We give three concrete application examples:

- We can replace the range proof showing that $b - a \in [0, B]$ (generated in CreateTransferTx) with a $\mathsf{Sharp}_{\mathsf{GS}}$ proof. Initially, the balance is 0. A funding transaction keeps the balance short (as the total number of coins are limited in Zether and the funded amount is public) and non-negative. Further, a range proof with non-relaxed soundness (for example Bulletproofs) ensures that the transfer amount $a \in [0, B]$ which implies that $b-a \in [-B, B]$, as the balance is non-negative. Thus, $\mathsf{Sharp}_{\mathsf{GS}}$ guarantees that $b-a \in [0, B]$. So by induction, the transfer transaction retains the invariant of a non-negative balance. As Bulletproofs allow batch proofs with tiny size overhead, the total size would increase to (at most) double. However, the verification cost of Bulletproofs scales linearly, so we expect the verification time to decrease significantly.

- Zether restricts coin withdrawal quite heavily as only the entire balance of an account can be withdrawn (see CreateBurnTx). When adding the functionality of partial burns to Zether, $\mathsf{Sharp}_{\mathsf{GS}}$ can be applied. A partial burn removes an amount a (specified by the user) from the user's balance b encrypted in (C, R). Now, the user has to prove that $b - a \in [0, B]$ (after subtracting a from the encrypted balance homomorphically). Zether ensures that $b \in [0, B]$ and as a is public, the smart contract can verify whether $a \in [0, B]$. Thus, $b - a \in [-B, B]$ is short and $\mathsf{Sharp}_{\mathsf{GS}}$ shows that $b - a \in [0, B]$ as desired. In this setting, our range proofs are more efficient than Bulletproofs.

- Zether does only allow private transfers (where the amount a is hidden). For some transactions, a user might want to reveal the amount a in exchange for a more efficient transfer. In this case, the user still needs to show that his balance b remains non-negative after the transfer, i.e. $b - a \in [0, B]$. Again, since the balance before the transfer is short and the transfer amount is public, $\mathsf{Sharp}_{\mathsf{GS}}$ guarantees soundness and is more efficient than using Bulletproofs.

Note that calculation of gas cost for cryptographic operation is very coarse, e.g. there is no difference between full-size or small exponentiations [Bün+20]. Unfortunately, Sharp proofs derive efficiency by computing with short exponents.

Leveraging RSA Without Trusted Setup. In transaction systems with trusted setup, e.g. Zcash [Zca], we can replace all range proofs in Zether with $\mathsf{Sharp}_{\mathsf{RSA}}$. Without trusted setup, we can use class groups and the four square decomposition. Now, $\mathsf{Sharp}_{\mathsf{CL}}$ guarantees membership in $[0, B]$ (for rational representatives) and we can perform an arbitrary number of additions as a bound B on the total sum of each balance is known (cf. appendix B). Unfortunately, dishonest users can still introduce fractions in the balance of other users (cf. section 7.3, denial of service). So we need to restrict dishonest users such that they cannot transfer fractional amounts. Note that for security, it is sufficient to restrict fractional transfers to *honest* users, as fractional transfers between dishonest parties cannot create coins and the system correctness is not harmed (for honest users).

We sketch how to leverage the properties of RSA groups *without* trusted setup to resolve this problem. First, we identify each user by a RSA modulus n that is privately generated,

in addition to the ElGamal public key Y. [14] Then, the users perform each range proof with $\mathsf{Sharp}_{\mathsf{CL}}$, i.e. Sharp augmented with a class group element (see appendix C). In addition, the proof that $a \geq 0$ is further augmented with an RSA group element with modulus n, where n is the public key of the receiver. The latter enforces the opening of a to be an integer, if n was setup honestly. (Note that one class group element is sufficient for augmenting both range proofs via MPed.) We have to distinguish two cases to analyze the security of a transfer:

- If n is a RSA modulus and the receiver's factorization is not known to the prover, the RSA group element guarantees that $a \geq 0$ is an *integer* and in the bounds $[0, B]$, as the square decomposition holds. Though the sender's balance might be fractional, $b - a \geq 0$ is guaranteed to be non-negative by the other range proof.
- If the receiver's factorization is known to the prover (or n is not a RSA modulus), the RSA group element provides no additional security guarantees. But as we augment the range proof also with a class group element, the transaction cannot generate coins, only introduce factions in the receiver's balance.

Thus, honest users profit from all standard security guarantees, whereas dishonest users can only interact with honest users while their balance is non-fractional. Note that if n and the MPed parameters in the RSA group are setup maliciously, the RSA commitment could leak information about a, but this only impacts transactions to dishonest users.

For 128 bits of security, class groups require a discriminant of size 1827 bits and RSA groups a modulus size of 3072 bits [BJS10; Thy+21]. Further, the representation of class group elements can be compressed to 3/4 the size of the discriminant [DGS21]. Thus, each transaction requires 555 Bytes of communication for both the additional RSA and class group element, in addition to the bare $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ range proof in a 256-bit group.

Unfortunately, this induces a slight communication overhead. Thus, we do not expect this approach to improve efficiency. Nevertheless, it shows that the square decomposition approach can be applied in a wide range of applications and that the properties of RSA groups can be leveraged, *without* trusted setup.

Acknowledgements

This work was supported by ANR SCENE and PEPR SecureCompute, and by funding from the topic Engineering Secure Systems of the Helmholtz Association (HGF) and by KASTEL Security Research Labs.

References

[Abr+22] Damiano Abram, Ivan Damgård, Claudio Orlandi, and Peter Scholl. *An Algebraic Framework for Silent Preprocessing with Trustless Setup and Active Security*. Cryptology ePrint Archive, Paper 2022/363. `https://eprint.iacr.org/2022/363`. 2022. URL: `https://eprint.iacr.org/2022/363`.

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. "A Compressed Σ-Protocol Theory for Lattices". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 549–579.

[14] We could also use n as public key for Paillier encryption [Pai99] instead of ElGamal. This allows choosing the bound B on the maximal number of coins in the system beyond 2^{32}. Unfortunately, the message space is of *known* order n, so the encryption to the balance cannot be used as augmenting group element. (Further, one needs to ensure that the Paillier encryption is binding, even with a malicious setup.)

[AFK21] Thomas Attema, Serge Fehr, and Michael Klooß. *Fiat-Shamir Transformation of Multi-Round Interactive Proofs*. Cryptology ePrint Archive, Report 2021/1377. `https://eprint.iacr.org/2021/1377`. 2021.

[AL21] Martin R. Albrecht and Russell W. F. Lai. "Subtractive Sets over Cyclotomic Rings - Limits of Schnorr-Like Arguments over Lattices". In: *CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. LNCS. Virtual Event: Springer, Heidelberg, Aug. 2021, pp. 519–548.

[Bau+18a] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. "Sub-linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits". In: *CRYPTO 2018, Part II*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10992. LNCS. Springer, Heidelberg, Aug. 2018, pp. 669–699.

[Bau+18b] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. "More Efficient Commitments from Structured Lattice Assumptions". In: *SCN 18*. Ed. by Dario Catalano and Roberto De Prisco. Vol. 11035. LNCS. Springer, Heidelberg, Sept. 2018, pp. 368–385.

[Ben+15] Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. "Efficient Zero-Knowledge Proofs for Commitments from Learning with Errors over Rings". In: *ESORICS 2015, Part I*. Ed. by Günther Pernul, Peter Y. A. Ryan, and Edgar R. Weippl. Vol. 9326. LNCS. Springer, Heidelberg, Sept. 2015, pp. 305–325.

[BFS20] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. "Transparent SNARKs from DARK Compilers". In: *EUROCRYPT 2020, Part I*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12105. LNCS. Springer, Heidelberg, May 2020, pp. 677–706.

[BG10] Zvika Brakerski and Shafi Goldwasser. "Circular and Leakage Resilient Public-Key Encryption under Subgroup Indistinguishability - (or: Quadratic Residuosity Strikes Back)". In: *CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. LNCS. Springer, Heidelberg, Aug. 2010, pp. 1–20.

[BG18] Jonathan Bootle and Jens Groth. "Efficient Batch Zero-Knowledge Arguments for Low Degree Polynomials". In: *PKC 2018, Part II*. Ed. by Michel Abdalla and Ricardo Dahab. Vol. 10770. LNCS. Springer, Heidelberg, Mar. 2018, pp. 561–588.

[BJS10] Jean-Franγ cois Biasse, Michael J. Jacobson, and Alan K. Silvester. "Security Estimates for Quadratic Field Based Cryptosystems". In: *ACISP 10*. Ed. by Ron Steinfeld and Philip Hawkes. Vol. 6168. LNCS. Springer, Heidelberg, July 2010, pp. 233–247.

[BL13] Foteini Baldimtsi and Anna Lysyanskaya. "Anonymous credentials light". In: *ACM CCS 2013*. Ed. by Ahmad-Reza Sadeghi, Virgil D. Gligor, and Moti Yung. ACM Press, Nov. 2013, pp. 1087–1098.

[Blö+19] Johannes Blömer, Jan Bobolz, Denis Diemert, and Fabian Eidens. "Updatable Anonymous Credentials and Applications to Incentive Systems". In: *ACM CCS 2019*. Ed. by Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz. ACM Press, Nov. 2019, pp. 1671–1685.

[Bob+20] Jan Bobolz, Fabian Eidens, Stephan Krenn, Daniel Slamanig, and Christoph Striecks. "Privacy-Preserving Incentive Systems with Highly Efficient Point-Collection". In: *ASIACCS 20*. Ed. by Hung-Min Sun, Shiuh-Pyng Shieh, Guofei Gu, and Giuseppe Ateniese. ACM Press, Oct. 2020, pp. 319–333.

[Boo+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. "Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting". In: *EUROCRYPT 2016, Part II*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. LNCS. Springer, Heidelberg, May 2016, pp. 327–357.

[Bou00] Fabrice Boudot. "Efficient Proofs that a Committed Number Lies in an Interval". In: *EUROCRYPT 2000*. Ed. by Bart Preneel. Vol. 1807. LNCS. Springer, Heidelberg, May 2000, pp. 431–444.

[Bra00] Stefan Brands. *Rethinking public key infrastructures and digital certificates: building in privacy*. Mit Press, 2000.

[Bün+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. "Bulletproofs: Short Proofs for Confidential Transactions and More". In: *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2018, pp. 315–334.

[Bün+20] Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. "Zether: Towards Privacy in a Smart Contract World". In: *FC 2020*. Ed. by Joseph Bonneau and Nadia Heninger. Vol. 12059. LNCS. Springer, Heidelberg, Feb. 2020, pp. 423–443.

[CC18] Pyrros Chaidos and Geoffroy Couteau. "Efficient Designated-Verifier Non-interactive Zero-Knowledge Proofs of Knowledge". In: *EUROCRYPT 2018, Part III*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10822. LNCS. Springer, Heidelberg, 2018, pp. 193–221.

[CCs08] Jan Camenisch, Rafik Chaabouni, and abhi shelat. "Efficient Protocols for Set Membership and Range Proofs". In: *ASIACRYPT 2008*. Ed. by Josef Pieprzyk. Vol. 5350. LNCS. Springer, Heidelberg, Dec. 2008, pp. 234–252.

[CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. "A Secure and Optimally Efficient Multi-Authority Election Scheme". In: *EUROCRYPT'97*. Ed. by Walter Fumy. Vol. 1233. LNCS. Springer, Heidelberg, May 1997, pp. 103–118.

[Cha90] David Chaum. "Showing Credentials without Identification Transferring Signatures between Unconditionally Unlinkable Pseudonyms". In: *AUSCRYPT'90*. Ed. by Jennifer Seberry and Josef Pieprzyk. Vol. 453. LNCS. Springer, Heidelberg, Jan. 1990, pp. 246–264.

[CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. "Compact E-Cash". In: *EUROCRYPT 2005*. Ed. by Ronald Cramer. Vol. 3494. LNCS. Springer, Heidelberg, May 2005, pp. 302–321.

[CK18] Henry Corrigan-Gibbs and Dmitry Kogan. "The Discrete-Logarithm Problem with Preprocessing". In: *EUROCRYPT 2018, Part II*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10821. LNCS. Springer, Heidelberg, 2018, pp. 415–447.

[CL01] Jan Camenisch and Anna Lysyanskaya. "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation". In: *EUROCRYPT 2001*. Ed. by Birgit Pfitzmann. Vol. 2045. LNCS. Springer, Heidelberg, May 2001, pp. 93–118.

[CL03] Jan Camenisch and Anna Lysyanskaya. "A Signature Scheme with Efficient Protocols". In: *SCN 02*. Ed. by Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano. Vol. 2576. LNCS. Springer, Heidelberg, Sept. 2003, pp. 268–289.

[CL15] Guilhem Castagnos and Fabien Laguillaumie. *Linearly Homomorphic Encryption from DDH*. Cryptology ePrint Archive, Report 2015/047. `https://eprint.iacr.org/2015/047`. 2015.

[CMZ14] Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. "Algebraic MACs and Keyed-Verification Anonymous Credentials". In: *ACM CCS 2014*. Ed. by Gail-Joon Ahn, Moti Yung, and Ninghui Li. ACM Press, Nov. 2014, pp. 1205–1216.

[Cou+21a] Geoffroy Couteau, Michael Klooß, Huang Lin, and Michael Reichle. *Efficient Range Proofs with Transparent Setup from Bounded Integer Commitments*. EUROCRYPT. 2021.

[Cou+21b] Geoffroy Couteau, Michael Klooß, Huang Lin, and Michael Reichle. "Efficient Range Proofs with Transparent Setup from Bounded Integer Commitments". In: *EUROCRYPT 2021, Part III*. Ed. by Anne Canteaut and Franγcois-Xavier Standaert. Vol. 12698. LNCS. Springer, Heidelberg, Oct. 2021, pp. 247–277.

[CPP17] Geoffroy Couteau, Thomas Peters, and David Pointcheval. "Removing the Strong RSA Assumption from Arguments over the Integers". In: *EUROCRYPT 2017, Part II*. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10211. LNCS. Springer, Heidelberg, 2017, pp. 321–350.

[CR19] Geoffroy Couteau and Michael Reichle. "Non-interactive Keyed-Verification Anonymous Credentials". In: *PKC 2019, Part I*. Ed. by Dongdai Lin and Kazue Sako. Vol. 11442. LNCS. Springer, Heidelberg, Apr. 2019, pp. 66–96.

[DF02] Ivan Damgård and Eiichiro Fujisaki. "A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order". In: *ASIACRYPT 2002*. Ed. by Yuliang Zheng. Vol. 2501. LNCS. Springer, Heidelberg, Dec. 2002, pp. 125–142.

[DGS21] Samuel Dobson, Steven Galbraith, and Benjamin Smith. "Trustless unknown-order groups". In: 2021. URL: `https://eprint.iacr.org/2020/196`.

[ElG84] Taher ElGamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". In: *CRYPTO'84*. Ed. by G. R. Blakley and David Chaum. Vol. 196. LNCS. Springer, Heidelberg, Aug. 1984, pp. 10–18.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. "Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations". In: *CRYPTO'97*. Ed. by Burton S. Kaliski Jr. Vol. 1294. LNCS. Springer, Heidelberg, Aug. 1997, pp. 16–30.

[FSW03] Pierre-Alain Fouque, Jacques Stern, and Jan-Geert Wackers. "CryptoComputing with Rationals". In: *FC 2002*. Ed. by Matt Blaze. Vol. 2357. LNCS. Springer, Heidelberg, Mar. 2003, pp. 136–146.

[GI08] Jens Groth and Yuval Ishai. "Sub-linear Zero-Knowledge Argument for Correctness of a Shuffle". In: *EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. LNCS. Springer, Heidelberg, Apr. 2008, pp. 379–396.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof Systems". In: *SIAM Journal on Computing* 18.1 (1989), pp. 186–208.

[Gro05] Jens Groth. "Non-interactive Zero-Knowledge Arguments for Voting". In: *ACNS 05*. Ed. by John Ioannidis, Angelos Keromytis, and Moti Yung. Vol. 3531. LNCS. Springer, Heidelberg, June 2005, pp. 467–482.

[Gro11] Jens Groth. "Efficient Zero-Knowledge Arguments from Two-Tiered Homomorphic Commitments". In: *ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. LNCS. Springer, Heidelberg, Dec. 2011, pp. 431–448.

[Har+17] Gunnar Hartung, Max Hoffmann, Matthias Nagel, and Andy Rupp. "BBA+: Improving the Security and Applicability of Privacy-Preserving Point Collection". In: *ACM CCS 2017*. Ed. by Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu. ACM Press, 2017, pp. 1925–1942.

[HKR19] Max Hoffmann, Michael Klooß, and Andy Rupp. “Efficient Zero-Knowledge Arguments in the Discrete Log Setting, Revisited”. In: *ACM CCS 2019*. Ed. by Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz. ACM Press, Nov. 2019, pp. 2093–2110.

[HN04] Johan Håstad and Mats Näslund. “The security of all RSA and discrete log bits”. In: *J. ACM* 51.2 (2004), pp. 187–230. URL: https://doi.org/10.1145/972639.972642.

[Hof+20] Max Hoffmann, Michael Klooß, Markus Raiber, and Andy Rupp. “Black-Box Wallets: Fast Anonymous Two-Way Payments for Constrained Devices”. In: *Proc. Priv. Enhancing Technol.* 2020.1 (2020), pp. 165–194. URL: https://doi.org/10.2478/popets-2020-0010.

[JR16] Tibor Jager and Andy Rupp. “Black-Box Accumulation: Collecting Incentives in a Privacy-Preserving Way”. In: *Proc. Priv. Enhancing Technol.* 2016.3 (2016), pp. 62–82. URL: https://doi.org/10.1515/popets-2016-0016.

[KK04] Takeshi Koshiba and Kaoru Kurosawa. “Short Exponent Diffie-Hellman Problems”. In: *PKC 2004*. Ed. by Feng Bao, Robert Deng, and Jianying Zhou. Vol. 2947. LNCS. Springer, Heidelberg, Mar. 2004, pp. 173–186.

[Lin03] Yehuda Lindell. “Parallel Coin-Tossing and Constant-Round Secure Two-Party Computation”. In: *Journal of Cryptology* 16.3 (June 2003), pp. 143–184.

[Lip03] Helger Lipmaa. “On Diophantine Complexity and Statistical Zero-Knowledge Arguments”. In: *ASIACRYPT 2003*. Ed. by Chi-Sung Laih. Vol. 2894. LNCS. Springer, Heidelberg, 2003, pp. 398–415.

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plancon. *Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General*. Cryptology ePrint Archive, Paper 2022/284. https://eprint.iacr.org/2022/284. 2022. URL: https://eprint.iacr.org/2022/284.

[LNS20] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. “Practical Lattice-Based Zero-Knowledge Proofs for Integer Relations”. In: *ACM CCS 2020*. Ed. by Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna. ACM Press, Nov. 2020, pp. 1051–1070.

[LW88] Douglas L. Long and Avi Wigderson. “The Discrete Logarithm Hides $O(\log n)$ Bits”. In: *SIAM Journal on Computing* 17.2 (1988), pp. 363–372.

[Lyu09] Vadim Lyubashevsky. “Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures”. In: *ASIACRYPT 2009*. Ed. by Mitsuru Matsui. Vol. 5912. LNCS. Springer, Heidelberg, Dec. 2009, pp. 598–616.

[Mon] *https://github.com/monero-project/monero*. Monero project.

[Pai99] Pascal Paillier. “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes”. In: *EUROCRYPT’99*. Ed. by Jacques Stern. Vol. 1592. LNCS. Springer, Heidelberg, May 1999, pp. 223–238.

[Ped92] Torben P. Pedersen. “Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing”. In: *CRYPTO’91*. Ed. by Joan Feigenbaum. Vol. 576. LNCS. Springer, Heidelberg, Aug. 1992, pp. 129–140.

[Per86] René Peralta. “Simultaneous Security of Bits in the Discrete Log”. In: *EUROCRYPT’85*. Ed. by Franz Pichler. Vol. 219. LNCS. Springer, Heidelberg, Apr. 1986, pp. 62–72.

[Pol78] John M Pollard. “Monte Carlo methods for index computation ($\mathsf{mod}\, p$)”. In: *Mathematics of computation* 32.143 (1978), pp. 918–924.

[PS19] Paul Pollack and Peter Schorn. “Dirichlet’s proof of the three-square theorem: An algorithmic perspective”. In: *Math. Comput.* 88.316 (2019), pp. 1007–1019. URL: https://doi.org/10.1090/mcom/3349.

[RS86] Michael O. Rabin and Jeffery O. Shallit. “Randomized Algorithms in Number Theory”. In: vol. 39. S1. 1986, S239–S256.

[RVH17] Sietse Ringers, Eric R. Verheul, and Jaap-Henk Hoepman. “An Efficient Self-blindable Attribute-Based Credential Scheme”. In: *FC 2017*. Ed. by Aggelos Kiayias. Vol. 10322. LNCS. Springer, Heidelberg, Apr. 2017, pp. 3–20.

[Sch98] Claus P. Schnorr. *Almost All Discrete Log Bits Are Simultaneously Secure*. Cryptology ePrint Archive, Report 1998/020. https://eprint.iacr.org/1998/020. 1998.

[Thy+21] Sri Aravinda Krishnan Thyagarajan, Guilhem Castagnos, Fabian Laguillaumie, and Giulio Malavolta. “Efficient CCA Timed Commitments in Class Groups”. In: *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2021, pp. 2663–2684.

[Val+19] Henry de Valence, Jack Grigg, George Tankersley, Filippo Valsorda, and Isis Lovecruft. *The ristretto255 group*. Tech. rep. IETF CFRG Internet Draft, 2019.

[vW96] Paul C. van Oorschot and Michael J. Wiener. “On Diffie-Hellman Key Agreement with Short Exponents”. In: *EUROCRYPT’96*. Ed. by Ueli M. Maurer. Vol. 1070. LNCS. Springer, Heidelberg, May 1996, pp. 332–343.

[Wik21] Douglas Wikström. *Special Soundness in the Random Oracle Model*. Cryptology ePrint Archive, Report 2021/1265. https://eprint.iacr.org/2021/1265. 2021.

[Wui18] Pieter Wuille. “libsecp256k1”. In: *URL: https://github. com/bitcoin/secp256k1* (2018).

[Zca] *https://github.com/zcash/zcash*. Zcash project.

A. Preliminaries Continued

We formalize the sketches from section 2.3. Some definitions are adapted from [Cou+21a].

A.1. Hash Functions

Definition A.1 (CRHF). Let $\mathsf{Hash}\colon \mathcal{K} \times \{0,1\}^* \mapsto \{0,1\}^\ell$ be a hash function. We call Hash a **collision-resistant** hash function (CRHF), if for all PPT adversaries $\mathscr{A}$ there exists a negligible function negl such that

$$\Pr\left[\begin{matrix} k \xleftarrow{\$} \mathcal{K}; (m_0, m_1) \leftarrow \mathscr{A}(1^\lambda, k): \\ m_0 \neq m_1 \wedge \mathsf{Hash}(k, m_0) = \mathsf{Hash}(k, m_1) \end{matrix}\right] \leq \mathrm{negl}(\lambda).$$

Recall that, due to generic birthday attacks, we need at least $l = 2\lambda$ output size for λ bits of security. Moreover, *keyed* hash functions are required to achieve collision-resistance against *non-uniform* adversaries, otherwise advice could contain collisions.

A.2. Commitments

A **(non-interactive) commitment scheme** Com allow committing to a message $m \in \mathcal{M}_{\mathrm{com}}$ obtaining a commitment $c \in \mathcal{C}_{\mathrm{com}}$ and opening information $r \in \mathcal{R}_{\mathrm{com}}$, where $\mathcal{M}_{\mathrm{com}}$, $\mathcal{C}_{\mathrm{com}}$, $\mathcal{R}_{\mathrm{com}}$ are message, commitment and opening (or randomness) space of Com, respectively. We now define the security properties formally.

Definition A.2 (Correctness of a Commitment Scheme). A commitment scheme Com is **correct**, if there exists a negligible function negl such that for any $\mathrm{ck} \xleftarrow{\$} \mathsf{Com.Setup}(1^\lambda)$, any message $m \in \mathcal{M}_{\mathrm{com}}$ and for $(c, d) \leftarrow \mathsf{Com.Commit}_{\mathrm{ck}}(m)$, it holds that $\mathsf{Com.Verify}_{\mathrm{ck}}(c, d, m) = 1 - \mathrm{negl}(\lambda)$.

Definition A.3 (Hiding Property of a Commitment Scheme). The advantage of an adversary $\mathscr{A}$ against a the hiding property of a commitment scheme Com is

$$\mathsf{Adv}^{\mathrm{hide}}_{\mathscr{A}}(\lambda) = \Pr\left[\begin{matrix} \mathrm{ck} \xleftarrow{\$} \mathsf{Com.Setup}(1^\lambda);\ b \xleftarrow{\$} \{0,1\}; \\ (m_0, m_1) \leftarrow \mathscr{A}(\mathrm{ck}); \\ (c, d) \leftarrow \mathsf{Com.Commit}_{\mathrm{ck}}(m_b); \\ b' \leftarrow \mathscr{A}(c)\colon b' = b \end{matrix}\right]$$

A commitment scheme Com is **hiding** if for any stateful PPT adversary $\mathscr{A}$, there exists a negligible function negl such that $\mathsf{Adv}^{\mathrm{hide}}_{\mathscr{A}}(\lambda) \leq \frac{1}{2} + \mathrm{negl}(\lambda)$.

Definition A.4 (Binding Property of a Commitment Scheme). The advantage of an adversary $\mathscr{A}$ against a the binding property of a commitment scheme Com is

$$\mathsf{Adv}^{\mathrm{hide}}_{\mathscr{A}}(\lambda) = \Pr\left[\begin{matrix} \mathrm{ck} \xleftarrow{\$} \mathsf{Com.Setup}(1^\lambda); \\ (c, d_0, d_1, m_0, m_1) \leftarrow \mathscr{A}(\mathrm{ck}) : \\ \mathsf{Com.Verify}_{\mathrm{ck}}(c, d, m_0) = 1 \\ \wedge\, \mathsf{Com.Verify}_{\mathrm{ck}}(c, d, m_1) = 1 \wedge m_0 \neq m_1 \end{matrix}\right]$$

A commitment scheme Com is **binding** if for any PPT adversaries $\mathscr{A}$, there exists a negligible function negl such that $\mathsf{Adv}^{\mathrm{bind}}_{\mathscr{A}}(\lambda) \leq \mathrm{negl}(\lambda)$.

A.3. Cryptographic Groups

Definition A.5 (S-Bounded DLSE and SEI). Consider a group $\mathbb{G}$. The S-bounded **discrete logarithm with short exponents** (DLSE) assumption holds if for all PPT $\mathscr{A}$ there is a negligible function negl such that

$$\Pr\left[G \xleftarrow{\$} \mathbb{G}; z \xleftarrow{\$} [0, S],\ z' \leftarrow \mathscr{A}(G, zG)\colon\ z = z'\right] \leq \mathrm{negl}(\lambda)$$

This probability defines the advantage $\mathsf{Adv}^{\mathrm{dlse}}_{\mathscr{A}}$ of $\mathscr{A}$ against DLSE.

The S-bounded **short exponent indistinguishability** (SEI) assumption holds if for all PPT $\mathscr{A}$ there is a negligible negl function such that

$$\begin{aligned}&\Pr\left[G \xleftarrow{\$} \mathbb{G}; z \xleftarrow{\$} [0, S] \colon \mathscr{A}(G, zG) = 1\right]\\ -&\Pr\left[G \xleftarrow{\$} \mathbb{G}; z \xleftarrow{\$} \mathbb{Z}_{\mathsf{ord}(G)} \colon \mathscr{A}(G, zG) = 1\right]\\ \leq&\,\mathrm{negl}(\lambda)\end{aligned}$$

This probability defines the advantage $\mathsf{Adv}^{\mathrm{sei}}_{\mathscr{A}}(\lambda)$ of $\mathscr{A}$ against SEI.

For groups of known order p, SEI holds unconditionally for $S = p - 1$. More generally, an unbounded adversary against SEI for $S = LU_{\mathsf{up}}$ has advantage at most $1/L$ in groups of unknown order (remark A.7), but relying on the SEI assumption for $S > U_{\mathsf{up}}$ is of little interest.

Note that SEI is a (long-standing) highly plausible assumption. Further, the DLSE and SEI assumption are known to be essentially equivalent in groups of known prime order with random generators [KK04], but a security loss is incurred in the reduction.

Remark A.6 (On SEI and DLSE). The study of the DLSE problem is essentially as old as that of the discrete logarithm problem itself: early works, when studying the discrete logarithm problem, typically considered the setting of bounded size exponents as well. See for example the seminal 1978 paper of Pollard [Pol78] and the "catching kangaroos" algorithm. It was also studied independently of the general DLOG problem, e.g. by van Oorschot and Wiener [vW96] (in 1996).

The SEI problem was first shown to be equivalent to the DLSE problem by Koshiba and Kurosawa [KK04] (in 2004). However, the roots of this equivalence are much older: the problem of distinguishing $\{g^x \mid x \xleftarrow{\$} \mathbb{Z}_p\}$ from $\{g^x \mid x \xleftarrow{\$} \text{short}\}$ is, in essence, the problem of predicting the most significant bits of x (e.g. if we define "short" as "smaller than p/2", then answering "is x short" is exactly answering "does the MSB of x equal 1"). The difficulty of extracting the individual bits of x (and its MSBs in particular) has been the subject of a long line of work, starting with Peralta [Per86] in 1985, and followed by Long and Wigderson [LW88] in 1988, Håstad and Näslund [HN04] in 1996, Schnorr [Sch98] in 1998, and many more.

The security loss between DLSE and SEI is t, where t is the bitlength of the short exponent (e.g. if the short exponents are 256 bits long, the reduction 'loses' 8 bits of security). However, no better attack is known on SEI compared to the DLSE, and it enjoys the same hardness in the generic group model. Hence, it is typically believed to achieve the same security level, without accounting for the reduction loss.

Groups of hidden order Following assumptions are relevant in groups of hidden order. Note that we still use additive notation, even if multiplicative notation is more common for RSA groups.

Remark A.7. In a cyclic group $\langle G\rangle$ of unknown order, a random group element be approximated via xG for $x \xleftarrow{\$} [0, \ldots, LU_{\mathsf{up}} - 1]$ and xG has statistical distance at most $1/L$ from a random group element. Indeed, at most $1/4L$ (due to [CL15]).

The ORD assumption ensures, that it is hard to find (a multiple of) the order of non-trivial elements.

Definition A.8 (ORD). The **order (ORD) assumption** holds for a given group $\mathbb{G}$ if for any PPT adversary $\mathscr{A}$, there is a negligible function negl, such that

$$\Pr\left[\begin{matrix}(W, \alpha) \leftarrow \mathscr{A}(\mathbb{G}); W \in \mathbb{G} \setminus \{0\};\\ 0 \neq |\alpha| < 2^{\mathsf{poly}(\lambda)} \colon \alpha W = 0\end{matrix}\right] \leq \mathrm{negl}(\lambda)$$

This probability defines the advantage $\mathsf{Adv}^{\mathrm{ORD}}_{\mathscr{A}}(\lambda)$ of $\mathscr{A}$ against ORD.

We use the adaption of ORD to the class group setting of [BFS20] with corrections from [Cou+21a]. It is believed to hold in suitable class groups of imaginary quadratic orders and the subgroup of quadratic residues QR_n in RSA groups. For 128 bits of security, class groups require a discriminant of size 1827 bits and RSA groups a modulus size of 3072 bits [BJS10; Thy+21]. Further, the representation of class group elements can be compressed to 3/4 the size of the discriminant [DGS21].

Definition A.9 (e-fROOT). The **e-fractional root (e-fROOT) assumption** holds for group $\mathbb{G}$ if for any PPT adversary $\mathscr{A}$, there is a negligible function negl, such that

$$\Pr\left[\begin{array}{c} G \xleftarrow{\$} \mathbb{G}; (\alpha, \beta, U) \leftarrow \mathscr{A}(\mathbb{G}, G); U \in \mathbb{G}; \\ 0 \neq |\alpha| < 2^{\mathsf{poly}(\lambda)} \in \mathbb{Z}; |\beta| < 2^{\mathsf{poly}(\lambda)} \in \mathbb{Z}: \\ \beta U = \alpha G \wedge \frac{\beta}{\gcd(\alpha,\beta)} \neq e^k \text{ for } k \in \mathbb{N} \end{array}\right] \leq \mathrm{negl}(\lambda)$$

This probability defines the advantage $\mathsf{Adv}^{e\text{-fROOT}}_{\mathscr{A}}(\lambda)$ of $\mathscr{A}$ against e-fROOT.

The e-strong RSA assumption is defined as e-fROOT but where $\alpha = 1$ must hold. [BFS20] define this and show that e-strong RSA and ORD imply e-fROOT. The e-fROOT assumption clearly implies e-strong RSA and almost implies ORD, except for elements W with $e^k W = 0$.

The 1-fROOT assumption is equivalent to the (usual) strong RSA assumption and believed to hold in QR_n. The 2-fROOT assumption is believed to hold in suitable class groups of imaginary quadratic orders.

Remark A.10. Let $\mathbb{G}$ be a group, let $G \in \mathbb{G}$, and let (α, β, W) with $\alpha G = \beta W$. Then:

1. The e-fROOT experiment is won with (α, β, W) iff $\frac{\alpha}{\beta} \notin \mathbb{Z}[1/e]$.
2. The ORD experiment is won if $d = \gcd(\alpha, \beta)$ (or more generally any divisor d of α and β), we have $\frac{\alpha}{d} G \neq \frac{\beta}{d} W$.

The SI assumption ensures that random elements in the subgroup $\langle G \rangle$ are indistinguishable from random elements in $\mathbb{G}$.

Definition A.11 (SI). The **subgroup indistinguishability (SI) assumption** holds for group $\mathbb{G}$ if for any PPT adversary $\mathscr{A}$, there is a negligible function negl, such that

$$\Pr\left[\begin{array}{c} G, H_0 \xleftarrow{\$} \mathbb{G}, H_1 \xleftarrow{\$} \langle G \rangle; \\ b \xleftarrow{\$} \{0,1\}, b' \leftarrow \mathscr{A}(\mathbb{G}, G, H_b): \\ b = b' \end{array}\right] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$$

The left hand side defines the advantage $\mathsf{Adv}^{\mathrm{si}}_{\mathscr{A}}(\lambda)$ of $\mathscr{A}$ against SI.

Again, we use the adaption from [BFS20] of the SI assumption introduced in [BG10]. It is believed to hold in QR_n or suitable class groups of imaginary quadratic orders.

Definition A.12 ((D, e, N)-relaxed DLOG-relation). Let $\mathbb{G}$ be a group, $D, e, N \in \mathbb{N}$, and $\vec{G} = (G_0, \ldots, G_N) \in \mathbb{G}^{N+1}$. Define the **$(D, e, N)$-relaxed DLOG relation w.r.t. $\vec{G}$** as

$$\mathsf{R}_{D,e,N}(\vec{G}) = \left\{ (C, d, \{m_i\}_{i=1}^N) \;\middle|\; \begin{array}{c} dC = \sum_{i=0}^N m_i G_i \wedge \exists i \colon \frac{m_i}{d} \notin \mathbb{Z}[1/e] \\ \wedge\, d \in [0, D] \wedge m_i \in \mathbb{Z} \end{array} \right\}$$

The advantage $\mathsf{Adv}^{\text{rel-dlog}}_{\mathbb{G},(D,e,N),\mathscr{A}}(\lambda)$ of $\mathscr{A}$ against the hardness of the (D, e, N)-relaxed DLOG-relation **with subgroup setup** (and without public coins), is defined as the following probability:

$$\Pr\left[\begin{array}{c} \mathbb{G} \leftarrow \mathsf{GenGrp}(1^\lambda); G_0 \xleftarrow{\$} \mathbb{G}; G_1, \ldots, G_N \xleftarrow{\$} \langle G_0 \rangle \\ (C, d, m_0, \ldots, m_N) \leftarrow \mathscr{A}(\mathbb{G}, G_0, \ldots, G_N): \\ (C, d, m_0, \ldots, m_N) \in \mathsf{R}_{D,e,N}(\vec{G}) \end{array}\right].$$

We say that finding (D, e, N)-relaxed DLOG-relations **with subgroup setup** is **hard** in $\mathbb{G}$, if for every PPT adversary, there exists a negligible function negl such that $\mathsf{Adv}^{\text{rel-dlog}}_{\mathbb{G},(D,e,N),\mathscr{A}}(\lambda) \leq \text{negl}(\lambda)$.

We define hardness **with random setup** analogously, except that $G_i \xleftarrow{\$} \mathbb{G}$ for all i (instead of $G_i \xleftarrow{\$} \langle G_0 \rangle$).

We abbreviate $(D, e, 1)$-relaxed by (D, e)-relaxed.

Viewing C as a commitment and $(G_0, \ldots, G_N)$ as a commitment key in definition A.12 (which is exactly how we use it), (D, e, N)-relaxed DLog-relation hardness roughly holds if it is not possibly to open C to anything but an element in $\mathbb{Z}[1/e]$ (where we neglect the condition that $d \leq D$). The choice $N = 1$ is the most important one for (D, e, N)-relaxed DLOG-relations, as it is required for our applications and (as we will see) is equivalent $N > 1$. The $(D, e, 0)$-relaxed DLOG-relation is a (presumably) slightly weaker assumption, but is implied under additional restrictions (or assumptions).

Lemma A.13. *Let $\mathbb{G}$ be a group and let $\mathscr{A}$ be an algorithm. Then for **hardness with subgroup setup**, we have following implications.*

1. *The (D, e, N)-relaxed DLOG-relation tightly implies the (D, e, n)-relaxed DLOG-relation for any $n \leq N$.*
2. *The $(D, e, 1)$-relaxed DLOG-relation tightly implies the (D, e, N)-relaxed DLOG-relation for $N \in \mathbb{N}_0$.*
3. *The e-fROOT assumption tightly implies hardness of $(D, e, 0)$-relaxed DLOG-relation. If $D = \infty$, the assumptions are equivalent.*
4. *If the order $|\mathbb{G}|$ has no prime factors smaller than or equal to D, then $(D, e, 0)$-relaxed DLOG-relation tightly implies the $(D, e, 1)$-relaxed DLOG-relation.*

*Under the SI assumption in $\mathbb{G}$, the claims also hold for **hardness with random setup**.*

Item 4 is the general formulation to be used with $C(\lambda)$-rough groups [DF02], i.e. groups which have no subgroups of order smaller than $C(\lambda)$. The proof of item 4 uses the argument from [Cou+21a] which is an adaption of [DF02]. For item 2, a standard randomization technique is used (which is also used to show the tight equivalence of DLOG and DLOG-relations). Items 1 and 3 are immediate and included for completeness.

Proof. To item 1: This is immediate: If $\mathscr{A}$ outputs $(C, d, m_0, \ldots, m_n)$, output $(C, d, m_0, \ldots, m_n, 0, \ldots, 0)$ to break (D, e, N)-relaxed DLOG-relation hardness with exactly the same success.

To item 2: For $N \leq 1$ this follows from the previous point. For $N \geq 2$, this follows with by borrowing randomization techniques from known prime order groups. Concretely, pick $\vec{r}_0, \vec{r}_1 \xleftarrow{\$} [0, 2N2^{\lambda} U_{\mathsf{up}}]^N$ and define the matrix

$$R = \begin{pmatrix} 1 & 0 \\ \vec{r}_0 & \vec{r}_1 \end{pmatrix} \in \mathbb{Z}^{(N+1)\times 2}$$

Let $(G'_0, \ldots, G'_N)^{\top} = R(G_0, G_1)^{\top}$. The reduction hands $(G'_0, \ldots, G'_N)$ to $\mathscr{A}$, which outputs $(d, C, m_0, \ldots, m_N)$. The reduction then returns $(d, C, (m_0, \ldots, m_N)R)$.

The success analysis will be information-theoretic. Let $K = \mathsf{ord}(G_0)$ be the order of the generated subgroup. Observe that information-theoretically $(\vec{r}_0, \vec{r}_1) \bmod K$ is almost uniform, namely the statistical distance to $\mathbb{Z}_K^N$ is at most $2^{-\lambda}$ to. Let g_1 be the DLOG of G_1 to G_0, i.e. $g_1 G_0 = G_1$. For simplicity, we now argue using the DLOGs, i.e. we argue over $\mathbb{Z}_K$ (mapping

G_0 to 1 and G_1 to g_1), and we argue as though $\vec{r}_0$ and $\vec{r}_1$ are uniform modulo K. Observe that $R(1,g_1)^\top$ and $(R+\vec{v}(-g_1,1))(1,g_1)^\top$ have the same distribution for any $\vec{v}\in\mathbb{Z}_K^{N+1}$, that is

$$R(1,g_1)^\top \sim (R+\vec{v}(-g_1,1))(1,g_1)^\top \tag{A.1}$$

In particular, this holds for uniformly random $\vec{v} \xleftarrow{\$} \mathbb{Z}_K^{N+1}$. Now consider $\mathscr{A}$'s output $(d,C,m_0,\ldots,m_N)$. Let $c=\mathsf{dlog}_{G_0}(C)$, i.e $c\cdot G=C$, and let $\vec{m}^\top=(m_0,\ldots,m_1)$. Then $dC=\sum_{i=0}^m m_iG_i=\vec{m}^\top R(G_0,G_1)^\top$ becomes $dc=\vec{m}^\top\vec{g}'=\vec{m}^\top R(1,g_1)$. Let $d'=d/\gcd(e^d,d)$, i.e. let $d'\mid d$ be the maximal factor of d which is coprime to e.

If $\mathscr{A}$ wins, then for some i we have $m_i/d\notin\mathbb{Z}[1/e]$. Moreover, following conditions are equivalent:

$$\begin{aligned} m_i/d\notin\mathbb{Z}[1/e] &\iff m_i/d'\notin\mathbb{Z}[1/e]\\ &\iff d'\nmid m_i\\ &\iff m_i\not\equiv_{d'}0\end{aligned}$$

Whenever $\vec{m}^\top R\not\equiv_{d'}0$ holds, then $(d,C,\vec{m}^\top R)$ is a $(D,e,1)$-relaxed DLOG-relation. Hence, we have to show that $\vec{m}^\top R\not\equiv_{d'}0$ holds with high probability. From the equivalence of distributions in eq. (A.1), we have

$$\begin{aligned} &\Pr[\vec{m}^\top R\equiv_{d'}\vec{0}]\\ =\;&\Pr[\vec{m}^\top(R+\vec{v}(-g_1,1))\equiv_{d'}\vec{0}]\\ \le\;&\max_{\vec{m}\not\equiv_{d'}0}\Pr[\vec{m}^\top\vec{v}(-g_1,1)\equiv_{d'}\vec{\mu}]\end{aligned}$$

where the initial probabilities go over R, m, v, and we used for the inequality that we can maximize over $\vec{m}$ and R and let $\vec{\mu}=-\vec{m}^\top R$. Looking only at the second component of the equation $\vec{m}^\top\vec{v}(-g_1,1)\equiv_{d'}\vec{\mu}$, namely, $\vec{m}^\top\vec{v}\equiv_{d'}\mu_2$, where $\vec{\mu}=(\mu_1,\mu_2)^\top$, suffices to upper-bound the probability.

Note that if $d'=1$, the adversary loses, so w.l.o.g. $d'\neq 1$. Let $p^k\mid d'$ be a prime power dividing d' such that $k\in\mathbb{N}$ is minimal with:

- for all $i=0,\ldots,N$: $m_i\not\equiv_{p^k}0$,
- for some $i=0,\ldots,N$: $m_i\equiv_{p^{k-1}}0$.

If no such p exists, then $m_i\equiv_{d'}0$ for all m_i, and again, $\mathscr{A}$ loses (because $m_i/d\in\mathbb{Z}[1/e]$ for all i). Thus, assume w.l.o.g. that such a prime p exists.

Now, we show

$$\Pr[\vec{m}^\top\vec{v}\equiv_{p^k}\mu_2]\le 1/p.$$

For this, intuitively, we consider the p-adic digits and concentrate on the k-th digit, and show that $\vec{m}^\top\vec{v}$ has almost uniformly random k-th digit. To do so, first note that (by assumption) all m_i lie in $p^{k-1}\mathbb{Z}$, and for some i^*, we have $m_i\notin p^k\mathbb{Z}$. In other words, we can divide all m_i by p^{k-1}, and then one element, namely m_{i^*}/p^{k-1}, is not divisible by p anymore. Thus, after dividing and taking the equations modulo p, we see that m_{i^*}/p^{k-1} is invertible in $\mathbb{Z}_p$. (Note that $m_i/p^{k-1}\ \mathsf{mod}\ p$ is exactly the k-th digit in basis p.) For the k-th digit of $\vec{m}^\top\vec{v}$, it is not hard to see that it is a uniformly linear combination of all the $m_i/p^{k-1}\ \mathsf{mod}\ p$, since $\vec{v}\ \mathsf{mod}\ p$ is uniform in $\mathbb{Z}_p^{N+1}$. But it is well-known that for $\vec{u}\xleftarrow{\$}\mathbb{Z}_p^{N+1}$ we have

$$\Pr[\vec{m}^\top\vec{u}\equiv_{p^k}\mu_2]=1/p$$

and easy to check that $\Pr[\vec{a}^\top\vec{u}\equiv_p z]=1/p$ if $\vec{u}\xleftarrow{\$}\mathbb{Z}_p^{N+1}$, $\vec{0}\neq\vec{a}\in\mathbb{Z}_p^{N+1}$, and $z\in\mathbb{Z}_p$.

Putting things together and accounting for the statistical distance of $2^{-\lambda}$ of the $\vec{r}_0$ and $\vec{r}_1$ from uniform over $\mathbb{Z}_K$, we have shown: If $\mathscr{A}$ does not lose, then we get

$$\Pr[\vec{m}^\top \vec{v} \neq \mu_2] \geq 1 - 1/p - 2^{-\lambda}$$

that is, with probability at least $1 - 1/p - \mathrm{negl}(\lambda)$ we find a non-trivial $(D, e, 1)$-relaxed relation. Thus, the claim follows.

To item 3: Observe that for $N = 0$, the $(D, e, 1)$-relaxed DLOG-relation specializes to hardness of finding (C, d, m) such that $dC = mG_0$ and $m/d \notin \mathbb{Z}[1/e]$, and $d \leq D$. Applying remark A.10 with $C = W$, $d = \beta$ and $m = \alpha$, and noting that $m = \alpha = 0$ breaks neither assumption, we immediately obtain the claimed equivalence if $D = \infty$, and a one-sided implication otherwise.

To item 4: Since it makes no difference in the proof, we directly show that (D, e, N)-relaxed implies $(D, e, 0)$-relaxed DLOG-relation hardness, *if* $|\mathbb{G}|$ *has no prime factor smaller or equal to* D. We setup $G_i = \rho_i G_0$ for $\rho_i \xleftarrow{\$} [0, N2^\lambda U_{\mathsf{up}}^2 - 1]$, where U_{up} is an upper bound on the group order of $\mathbb{G}$. By remark A.7, G_i is $1/N \cdot 2^{-\lambda} U_{\mathsf{up}}^{-1}$ close to a uniform element in $\langle G_0 \rangle$. By a union bound, $(G_1, \ldots, G_N)$ is $2^{-\lambda} U_{\mathsf{up}}^{-1}$ close to uniform in $\mathbb{G}$. Let $n \leq U_{\mathsf{up}}$ be any number with $\gcd(n, \mathsf{ord}(G_0)) = 1$. Since G_i information-theoretically only reveals $\rho_i \bmod \mathsf{ord}(G_0)$, we see that $(\rho_1, \ldots, \rho_N) \bmod n$ is $2^{-\lambda}$-close to uniform in $\mathbb{Z}_n$ (since $n \leq U_{\mathsf{up}}$). We apply this to $n = p^k/d$ later.

As in item 2, let (w.l.o.g.) $1 \neq d' \mid d$ be the maximal factor of d coprime to e, and let p be a prime and $k \in \mathbb{N}$ be minimal such that $m_i \equiv_{p^{k-1}} 0$ for all i, but $m_{i^*} \not\equiv_{p^k} 0$ for some $i^* \in \{0, \ldots, N\}$. Such p and i^* exist since $d' \neq 1$.

Using the setup, we have $dC = m_0 G_0 + \sum_{i=1}^N m_i G_i = (m_0 + \sum_{i=1}^N \rho_i m_i) G_0$. Let $m \coloneqq m_0 + \sum_{i=1}^N \rho_i m_i$. Observe that if $m \not\equiv_{p^k} 0$, then we have $dC = mG_0$ and $m \not\equiv_{d'} 0$, and hence $m/d \notin \mathbb{Z}[1/e]$ because d is not a power of e. Thus, we break e-fROOT. Now, we show that this happens with high probability.

Since we assumed that $m_{i^*} \not\equiv_{p^k} 0$, but $m_i \equiv_{p^{k-1}} 0$, we can argue almost as in item 2 that the linear combination $(\sum_{i=0}^N \rho_i m_i)/p^{k-1} \bmod p$ is zero with probability at most $1/p + \mathrm{negl}$. The only difference is that $\rho_0 = 1$. But it is easy to see that $z_0 + \sum_{i=1}^N u_i z_i \bmod p$ for arbitrary $z_i \in \mathbb{Z}$ and uniform $u_i \xleftarrow{\$} \mathbb{Z}_p$ is 0 with probability at most $1/p$, unless $z_i \bmod p = 0$ for all i.

Since $m \coloneqq m_0 + \sum_{i=1}^N \rho_i m_i$, we find that $m \bmod p^k \neq 0$ with probability at least $1 - 1/p - \mathrm{negl}$, and hence, whenever $\mathscr{A}$ wins, the reduction wins with probability at least $1/3$. Thus, the claim follows. □

A.3.1. Transparent setup and assumptions without invertible sampling

The above assumptions for cryptographic groups can *not* be directly used for transparent setup, because the adversary does not learn the random coins used to generate the group elements in the experiment. However, if the sampling procedure for uniform group elements is "invertible", one can find suitable coins given the resulting group elements, and thus, the assumptions are applicable also for transparent setup. In the following, we explain invertible sampling, and also generalize our assumptions to the setting where *no invertible sampling* is available.

Let Sample be an algorithm which given the group (or more generally some parameter) as input, samples an element, i.e. $G \leftarrow \mathsf{Sample}(1^\lambda, \mathbb{G}; r)$. A (overly restrictive, but sufficient) definition for **invertible sampling** is, that there exists an algorithm I, that given an element $G \in \mathrm{supp}(\mathsf{Sample}(1^\lambda, \mathbb{G}))$ samples random coins r, such that $\mathsf{Sample}(1^\lambda, \mathbb{G}; r)$ and $\mathsf{Sample}(1^\lambda, \mathbb{G}; I(1^\lambda, \mathbb{G}, \mathsf{Sample}(1^\lambda, \mathbb{G}; r), \rho))$ are statistically close. In other words, I can be used to "explain" any element G as being sampled via $\mathbb{G}$, by producing the correct coins (with the correct distribution), at least if the distribution of G is as in Sample. Often, invertible sampling is perfect, i.e. for any possible output G, I produces a uniform r with $G = \mathsf{Sample}(1^\lambda, \mathbb{G}; r)$.

If invertible sampling is available, one can exploit it to "program" a setup: Let Sample be the (transparent) algorithm used to sample (uniform) group elements. Instead of choosing uniform elements $G_i \xleftarrow{\$} \mathbb{G}$, i.e. $G_i \leftarrow \mathsf{Sample}(1^\lambda, \mathbb{G}; r_i)$, one can choose $G_0 \xleftarrow{\$} \mathbb{G}$ and $G_i = s_i G_0$, and then use invertible sampling to "explain" G_i as an honest choice $G_i \leftarrow \mathsf{Sample}(1^\lambda, \mathbb{G}, r_i)$, simply by using the invertible sampler for Sample to produce $r_i \leftarrow I(1^\lambda, \mathbb{G}, G_i)$.

Invertible sampling is known for many groups. Indeed, whenever group elements have unique bit-representations of (fixed) length ℓ and $2^\ell/|\mathbb{G}|$ is noticeable, invertible sampling is automatically possible. Say $2^\ell/|\mathbb{G}| \geq 1/100$, then Sample simply tries to interpret $r \xleftarrow{\$} \{0,1\}^\ell$ as a group element, and retries with fresh r until it succeeds (and if 100λ tries failed, Sample outputs $\perp$). Invertible sampling is clear: $I(1^\lambda, \mathbb{G}, G)$ first samples the try which is successful (by simulating Sample), and then outputs G in that try. From this naive procedure, we obtain invertible sampling in many cases of interest. For example, for $\mathbb{Z}_n^\times$ with typical n (e.g. n prime, or a product of few primes), it works as follows: Simply pick a random bitstring in $x \xleftarrow{\$} [0, 2^{\lceil \log(n) \rceil}] = \{0,1\}^{\lceil \log(n) \rceil}$, if $x \geq n$ retry, else if $\gcd(x, n) \neq 1$ retry, else output x. For this Sample procedure, the invertible sampler I works as follows: $I(1^\lambda, \mathbb{G}, G)$ runs $\mathsf{Sample}(1^\lambda, \mathbb{G}, G; r)$ and interprets $r = (r_i)_i$ where i is the randomness of the i-th trial. If the k-th trial accepts, I replaces r_k with the bitstring representing G and outputs this modified randomness.

All in all, invertible sampling is known for many groups. Unfortunately, invertible sampling is not known to be possible in class groups (at the time of writing). This was first pointed out by [Abr+22] w.r.t. CKLR proofs [Cou+21b], but also affects our setting. CKLR resolved the problem by relying on ElGamal commitments and the DXDH assumption [Abr+22] which sacrifices efficiency. We rely on novel assumptions which take this into account by providing the adversary with sampling randomness. We stress that, while these assumptions are stronger than their counterparts, like DXDH they are all very plausible.

In the following, we denote by Sample the sampling algorithm and write $(G, \rho) \xleftarrow{\$} \mathsf{Sample}(1^\lambda, \mathbb{G})$, i.e. we consider the random coins ρ as an output, partially adopting the notation of [Abr+22]. This is more convenient and more flexible as it allows us to model leakage other than the random coins as well. We modify our assumptions to account for leakage of ρ to the adversary. The modifications are straightforward, but we provide them for completeness. Changes are highlighted in red.

Definition A.14 (S-Bounded DLSE and SEI w.r.t. Sample). Consider a group $\mathbb{G}$. The S-bounded **discrete logarithm with short exponents** (DLSE) assumption w.r.t. Sample holds if for all PPT $\mathscr{A}$ there is a negligible function negl such that

$$\Pr\left[\begin{array}{c}(G,\rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathbb{G});\\ z \xleftarrow{\$} [0,S];\ z' \leftarrow \mathscr{A}(G,\rho,zG)\end{array} : z = z'\right] \leq \mathrm{negl}(\lambda)$$

The S-bounded **short exponent indistinguishability** (SEI) assumption w.r.t. Sample holds if for all PPT $\mathscr{A}$ there is a negligible negl function such that

$$\begin{aligned}&\Pr\left[(G,\rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathbb{G}); z \xleftarrow{\$} [0,S] : \mathscr{A}(G,\rho,zG) = 1\right]\\ -&\Pr\left[(G,\rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathbb{G}); z \xleftarrow{\$} \mathbb{Z}_{\mathsf{ord}(G)} : \mathscr{A}(G,\rho,zG) = 1\right]\\ \leq&\,\mathrm{negl}(\lambda)\end{aligned}$$

Definition A.15 (e-fROOT w.r.t. Sample). The **e-fractional root (e-fROOT) assumption w.r.t. Sample** holds for group $\mathbb{G}$ if for any PPT adversary $\mathscr{A}$, there is a negligible function negl, such that

$$\Pr\left[\begin{array}{c}(G,\rho) \leftarrow \mathsf{Sample}(1^\lambda, \mathbb{G}); (\alpha, \beta, U) \leftarrow \mathscr{A}(\mathbb{G}, G, \rho);\\ U \in \mathbb{G}; 0 \neq |\alpha| < 2^{\mathsf{poly}(\lambda)} \in \mathbb{Z}; |\beta| < 2^{\mathsf{poly}(\lambda)} \in \mathbb{Z}:\\ \beta U = \alpha G \wedge \frac{\beta}{\gcd(\alpha,\beta)} \neq e^k \text{ for } k \in \mathbb{N}\end{array}\right] \leq \mathrm{negl}(\lambda)$$

Definition A.16 (SI w.r.t. Sample). The **subgroup indistinguishability (SI) assumption w.r.t. Sample** holds for group $\mathbb{G}$ if for any PPT adversary $\mathscr{A}$, there is a negligible function negl, such that

$$\Pr\left[\begin{array}{c}(G,\rho)\leftarrow \mathsf{Sample}(1^\lambda,\mathbb{G}); H_0 \xleftarrow{\$} \mathbb{G}, H_1 \xleftarrow{\$} \langle G\rangle\,; \\ b \xleftarrow{\$} \{0,1\}, b' \leftarrow \mathscr{A}(\mathbb{G}, G, \rho, H_b): \\ b = b'\end{array}\right] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$$

Definition A.17 (Hard (D,e,N)-relaxed DLOG-relation (w.r.t. Sample)). Let $\mathbb{G}$ be a group, $D, e, N \in \mathbb{N}$, and $\vec{G} = (G_0, \ldots, G_N) \in \mathbb{G}^N$. The advantage $\mathsf{Adv}^{\text{rel-dlog}}_{\mathbb{G},(D,e,N),\mathscr{A}}(\lambda)$ of $\mathscr{A}$ in the advantage against hardness of (D,e,N)-relaxed DLOG-relation **w.r.t.** Sample, is defined as the following probability:

$$\Pr\left[\begin{array}{c}\forall i = 1,\ldots,N\colon (G_i,\rho_i)\leftarrow \mathsf{Sample}(1^\lambda,\mathbb{G}) \\ (C,d,m_0,\ldots,m_N)\leftarrow \mathscr{A}(\mathbb{G},G_0,\rho_0,\ldots,G_N,\rho_N)\colon \\ (C,d,m_0,\ldots,m_N)\in \mathsf{R}_{D,e,N}(\vec{G})\end{array}\right].$$

We say that finding (D,e,N)-relaxed DLOG-relations w.r.t. Sample is **hard**, if for every PPT adversary, there exists a negligible function negl such that $\mathsf{Adv}^{\text{rel-dlog}}_{\mathbb{G},(D,e,N),\mathscr{A}}(\lambda) \leq \mathrm{negl}(\lambda)$.

(Note: We do not define hardness with subgroup setup, as this case does not occur.)

A.4. Zero-Knowledge Proofs of Knowledge

We define zero-knowledge with setup GenCRS, which generates a common reference string (CRS) $\mathsf{crs} \leftarrow \mathsf{GenCRS}(\mathsf{pp})$. In this work, we only require an unstructured CRS, a.k.a. uniformly (common) random string (URS), which in practice is easier to come by than a structured reference string (SRS). Let R be a NP-relation over a set X defining a (pp-dependent) NP-language $\mathscr{L} = \{x \in X \mid \exists w : \mathsf{R}(\mathsf{pp}, x; w) = 1\}$. For simplicity, we suppress the dependency on pp when it is clear. A **proof system** for $\mathscr{L}$ is a protocol between a prover P and verifier V. We write $tr \leftarrow \langle \mathsf{P}(s), \mathsf{V}(t)\rangle$ for the *transcript* of an interaction where P (resp. V) has input s (resp. t) and *implicit inputs* $1^\lambda, \mathsf{pp}, \mathsf{crs}$. We write $b = \langle \mathsf{P}(s), \mathsf{V}(t)\rangle$ for the verifier's output b. A proof system is **public coin** if the verifier's messages are uniformly random and independent of the prover's messages, and the verifier outputs $b = \mathsf{Verify}(x, tr)$ for a PPT algorithm Verify.

Due to rejection sampling, our schemes have non-negligible correctness error.

Definition A.18 (Correctness). A proof system $(\mathsf{GenCRS}, \mathsf{P}, \mathsf{V})$ for $\mathscr{L}$ has **correctness error** γ_{err} if for every adversary $\mathscr{A}$

$$\Pr\left[\begin{array}{c}\mathsf{pp}\leftarrow \mathsf{GenPP}(1^\lambda); \mathsf{crs}\leftarrow \mathsf{GenCRS}(\mathsf{pp}); \\ (x,w)\leftarrow \mathscr{A}(\mathsf{pp},\mathsf{crs})\colon \\ \langle \mathsf{P}(\mathsf{pp},\mathsf{crs},x,w), \mathsf{V}(\mathsf{pp},\mathsf{crs},x)\rangle = 1\end{array}\right] \geq 1 - \gamma_{\mathsf{err}}(\lambda)$$

We call $(\mathsf{GenCRS}, \mathsf{P}, \mathsf{V})$ **correct** if $\gamma_{\mathsf{err}} = \mathrm{negl}$.

To separate (statistical) simulation and knowledge errors from hardness assumptions as much as possible, we define zero-knowledge and knowledge extraction by means of adversary advantages.

Definition A.19 ((Non-Abort) (S)HVZK). A simulator Sim for a public coin proof system $(\mathsf{GenCRS}, \mathsf{P}, \mathsf{V})$ for $\mathscr{L}$ is a PPT algorithm with input a statement x for which $(\mathsf{pp}, x, w) \in \mathsf{R}$ and

implicit inputs $1^\lambda, \mathsf{pp}, \mathsf{crs}$, and output a transcript tr. Let $\mathscr{A}$ be a stateful algorithm and let

$$\mathsf{Real}_{\mathscr{A}}(\lambda) = \Pr\left[\begin{array}{c}\mathsf{pp} \leftarrow \mathsf{GenPP}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{GenCRS}(\mathsf{pp}); \\ (x, w) \leftarrow \mathscr{A}(\mathsf{pp}, \mathsf{crs}); \\ tr \leftarrow \langle \mathsf{P}(\mathsf{pp}, \mathsf{crs}, x, w), \mathsf{V}(\mathsf{pp}, \mathsf{crs}, x)\rangle; \\ b \leftarrow \mathscr{A}(tr) \colon\ b \wedge \mathsf{R}(\mathsf{pp}, x; w) = 1\end{array}\right]$$

$$\mathsf{Ideal}_{\mathscr{A}}(\lambda) = \Pr\left[\begin{array}{c}\mathsf{pp} \leftarrow \mathsf{GenPP}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{GenCRS}(\mathsf{pp}); \\ (x, w) \leftarrow \mathscr{A}(\mathsf{pp}, \mathsf{crs}); tr \leftarrow \mathsf{Sim}(\mathsf{pp}, \mathsf{crs}, x); \\ b \leftarrow \mathscr{A}(tr) \colon\ b \wedge \mathsf{R}(\mathsf{pp}, x; w) = 1\end{array}\right]$$

Define the advantage of $\mathscr{A}$ by $\mathsf{Adv}^{\mathrm{hvzk}}_{\mathscr{A},\mathsf{P},\mathsf{V}}(\lambda) = \mathsf{Real}_{\mathscr{A}}(\lambda) - \mathsf{Ideal}_{\mathscr{A}}(\lambda)$. Then Sim (and by extension $(\mathsf{GenCRS}, \mathsf{P}, \mathsf{V})$) is **honest verifier zero-knowledge** with **simulation error** $\sigma_{\mathsf{err}} = \sigma_{\mathsf{err}}(\lambda)$, if for all PPT $\mathscr{A}$ there exists a negligible function negl such that $\mathsf{Adv}^{\mathrm{hvzk}}_{\mathscr{A},\mathsf{P},\mathsf{V}} \leq \sigma_{\mathsf{err}} + \mathrm{negl}$.

If Sim first samples all verifier challenges and then proceeds with the simulation, i.e. if Sim could take the challenges as additional input, it is a ***special*** *honest verifier zero-knowledge (SHVZK)* simulator.

The simulator is **non-abort** (S)HVZK, if it satisfies the weaker requirement, that simulated transcripts and real non-aborting transcripts are indistinguishable. Formally, using the modified $\mathsf{Real}_{\mathscr{A}}$, where the transcript tr is replaced by $\perp$ if the honest prover aborts, to define the advantage $\mathsf{Adv}^{\text{na-hvzk}}_{\mathscr{A},\mathsf{P},\mathsf{V}}$.

Remark A.20. Our protocols are only non-abort SHVZK. If "standard" (S)HVZK is needed, it can be obtained via well-known transformations. For example, via committing to those messages which, in case of failed masking, the simulator could not compute backwards. If these messages have enough entropy, suitable hashing (which is collision resistant and hides high-entropy preimages) suffices.

Definition A.21 (Knowledge Error). Let $(\mathsf{GenCRS}, \mathsf{P}, \mathsf{V})$ be a public coin proof system for $\mathscr{L}$. Let Ext be an *expected* polynomial time oracle algorithm (with oracle steps counted as one step) with implicit inputs $1^\lambda, \mathsf{pp}, \mathsf{crs}$. Let $\mathscr{A}$ be a probabilistic algorithm and P^* be a deterministic algorithm.

$$\mathsf{Real}_{\mathscr{A}}(\lambda) = \Pr\left[\begin{array}{c}\mathsf{pp} \leftarrow \mathsf{GenPP}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{GenCRS}(\mathsf{pp}); \\ (x, s) \leftarrow \mathscr{A}(\mathsf{pp}, \mathsf{crs}); \\ tr \leftarrow \langle \mathsf{P}^*(x, s), \mathsf{V}(x)\rangle \colon\ \mathsf{Verify}(x, tr) = 1\end{array}\right]$$

$$\mathsf{Ideal}_{\mathscr{A}}(\lambda) = \Pr\left[\begin{array}{c}\mathsf{pp} \leftarrow \mathsf{GenPP}(1^\lambda); \mathsf{crs} \leftarrow \mathsf{GenCRS}(\mathsf{pp}); \\ (x, s) \leftarrow \mathscr{A}(\mathsf{pp}, \mathsf{crs}); (tr, w) \leftarrow \mathsf{Ext}^{\mathsf{P}^*(x,s)} \colon \\ \mathsf{Verify}(x, tr) = 1 \wedge \mathsf{R}(\mathsf{pp}, x; w) = 1\end{array}\right]$$

W.l.o.g. Ext sets $w = \perp$ if $\mathsf{Verify}(x, tr) \neq 1$. The advantage of $(\mathscr{A}, \mathsf{P}^*)$ is $\mathsf{Adv}^{\mathrm{ke}}_{\mathscr{A},\mathsf{P}^*,\mathsf{V}}(\lambda) = \mathsf{Real}_{\mathscr{A}}(\lambda) - \mathsf{Ideal}_{\mathscr{A}}(\lambda)$. A proof system has **knowledge error** κ_{err}, if for any PPT pair $(\mathscr{A}, \mathsf{P}^*)$, there exists a negligible function negl such that $\mathsf{Adv}^{\mathrm{ke}}_{\mathscr{A},\mathsf{P}^*,\mathsf{V}} \leq \kappa_{\mathsf{err}} + \mathrm{negl}$.

In practice, one wants not only knowledge soundness, but also the ability to continue the simulation, which is called witness-extended emulation [Lin03; GI08]. Since all of our extractors are black-box and obtain an initial transcript by emulating an honest verifier, they trivially have perfect emulation by outputting that transcript (whether or not the extraction succeeds is a different question, and must be separated if one wants to work with non-negligible knowledge error).

The crucial components of our proof systems are either k-special sound, or similar techniques are applied. That is, given k "related transcripts", one can reconstruct a witness. The proof systems are not always k-special sound in a strict sense. Indeed, $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ is a 5-move protocol (which does not satisfy the tree-of-transcripts generalization of special soundness either).

Remark A.22 (Getting Related Transcripts). We can w.l.o.g. consider deterministic adversaries for knowledge extraction (due to linearity of expectation). For these, it is well known how to obtain related transcripts, in *expected* polynomial time. It is obvious for 3-move protocols: Just rewind and try fresh challenges. In expected constant rewinds, a second accepting challenge γ will be found. The probability that γ was not encountered before is at least $1/k$ (or 1 if sampling without replacement). Since $k = \mathsf{poly}$, k (distinct) transcript are found in expected polynomial time. Thus, we can assume w.l.o.g. that we have k transcripts. For more moves, similar arguments work, see e.g. [Boo+16; ACK21].

B. Further Remarks on Sharp's Soundness

The Sharp family satisfies correctness for short integers x in $\mathbb{Z}_p$, while it guarantees relaxed soundness, namely range membership of the rational representative $[x]_{\mathbb{Q}}$. Here, we discuss the behaviour of rational representatives and the soundness guarantees of (the variants of) Sharp from the perspective of possible use-cases.

B.1. Arithmetic Behaviour of $\mathbb{Q}_{M,D}$

Let $x_i, c \in \mathbb{Z}_p$ for $i \in [1, \ell]$. The usual integer representatives x_i behave very simply. Namely additions $x_1 + x_2$ and multiplications with constant $c \cdot x_1$ (and general multiplications $x_1 \cdot x_2$) work on representatives as long as it is ensured that no wraparound happens, i.e. the result is within $[-\frac{p-1}{2}, \frac{p-1}{2}]$. For example, for ℓ additions

$$\mathbb{Z} \ni \sum_{i=1}^{\ell} x_i = (\sum_{i=1}^{\ell} x_i) \bmod p \tag{B.1}$$

if $\ell M < p/2$ and each x_i is bound by M, everything works well. Similar claims hold for multiplications (with a constant $c \in \mathbb{Z}$). If there is a bound B on the total sum of values x_i, eq. (B.1) holds if $B < p/2$ and independently of the total number of additions ℓ.

B.1.1. Overflow Conditions for $\mathbb{Q}_{M,D}$.

For rational representatives, overflows behaviour effectively boils down to computations with fractions. Let $x_i \in \mathbb{Z}_p$ and let $[x_i]_{\mathbb{Q}_{M,D}} = n_i/d_i \in \mathbb{Q}_{M,D}$ be the rational representative of x_i for $i \in [1, \ell]$. The sum $x_1 + x_2$ has rational representative $(n_1 d_2 + n_2 d_1)/(d_1 d_1) \in \mathbb{Q}_{2MD,D^2}$ and similarly for multiplications (with constants). Note that $\mathbb{Q}_{M,D}$ is "two-dimensional" in the sense that M and D are independent (but must satisfy $MD < p/2$), If $\mathbb{Q}_{M,D} \subseteq \mathbb{Q}_{M',D'}$ then representatives will coincide, but in general (e.g. if $M < M'$ but $D > D'$) $\mathbb{Q}_{M,D}$ and $\mathbb{Q}_{M',D'}$ representatives have no obvious relation. In analogy to the summation example (eq. (B.1)), we can require $x_i \in \mathbb{Q}_{M,D}$ for $\ell M D^{2\ell-1} < p/2$ in order for

$$\sum_{i=1}^{\ell} [x_i]_{\mathbb{Q}_{M,D}} = [(\sum_{i=1}^{\ell} x_i) \bmod p]_{\mathbb{Q}_{M',D'}} \tag{B.2}$$

to hold in $\mathbb{Q}_{M',D'}$, where $M' \leq \ell M D^{\ell-1}$, $D' \leq D^{\ell}$. Similar claims hold for multiplication. Addition and multiplication of small integers $c \in \mathbb{Z}$ with $|c| < C$ behaves well: $c \cdot x_1 \in \mathbb{Q}_{CM,D}$.

Remark B.1. The potentially rapid growth of numerator and denominator under additions is one of the main sources of trouble when using rational representatives and relaxed soundness guarantees. Hence, they are less "friendly" in homomorphic operations on commitments.

B.1.2. Overflow Conditions in Hidden Order Groups.

With groups of hidden order, we show that the denominator is of the form $d_i = e^k \leq D$, cf. see appendix C. In particular, a sum $x_1 + x_2$ now lies in $\mathbb{Q}_{2MD,D}$, i.e. the maximal possible denominator D is unchanged — we prevented the growth of D. The requirement for eq. (B.2) to hold improves to ℓMD^2. When we further know a bound B on the sum of all numerators, the requirement becomes BD^2 and is independent of ℓ.

B.2. Remark on the Square Decomposition

We recall that the three square decomposition only shows relaxed range membership for fractions (unless one rounds to integers [Cou+21a] or has prior knowledge (section 6.3)).

Lemma B.2 (Three Squares for Fractions)**.** *Let $B \geq 1$, $x \in \mathbb{Q}$ and $\{x_i\}_{i=1}^3 \in \mathbb{Q}$ such that $1 + 4x(B - x) = \sum_{i=1}^3 x_i^2$. It holds that $x \in [-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}}$.*

To show exact range membership for fractions, we can use the four square decomposition.

Lemma B.3 (Four Squares for Fractions)**.** *Let $B \geq 1$, $x \in \mathbb{Q}$ and $\{x_i\}_{i=1..4} \in \mathbb{Q}$ and $B \in \mathbb{N}$. Further, let $x(B - x) = \sum_{i=1}^4 x^2$. Then it holds that $x \in [0, B]_{\mathbb{Q}}$.*

Both decompositions can be calculated efficiently [RS86; PS19]. As the four square decomposition increases the communication (since we have to open an additional committed integer x_4), we present our range proofs using the three square decomposition. Replacing it with the four square decomposition leads to range proofs that guarantee exact range membership for the rational representative. Alternatively, if $\Gamma < 4B$ is ensured (e.g. by, if necessary, trading challenge size for repetitions), our soundness claims ensure that denominators $d \geq 4B$ violate soundness, i.e. $[0, B]_{\mathbb{Q}_{K',\Gamma}} = [-\frac{1}{4B}, B + \frac{1}{B}]_{\mathbb{Q}} \cap \mathbb{Q}_{K',\Gamma}$.

C. Augmented Soundness

We show how to "augment" a range proof with one hidden order group element in order to improve the soundness guarantee. The hidden order group can be instantiated with: (1) a class group for better additive homomorphic guarantees (cf. appendix B.1.2) or (2) a RSA group for standard soundness with trusted setup.

C.1. Proof of Short Opening

Both the simple PoSO used in $\mathsf{Sharp}_{\mathsf{GS}}$ and the Batch-PoSO used in $\mathsf{Sharp}_{\mathsf{SO}}^{\mathsf{Po}}$ only ensure that the (committed) values are short as *fractions*, i.e. lie in $\mathbb{Q}_{M,D}$ for suitable M and D. It is easy to see, that these PoSOs also work over (commitments in) hidden order groups, as they are ignorant of the group order. Importantly, hidden order groups allow to mitigate the problems with homomorphic computations of fractions to some extent. Thus, we can achieve better soundness guarantees. Namely, under the hardness of (Γ, e)-relaxed DLOG relations, denominators d of an extracted witness $x = m/d$ must be of the form $d = e^k$, for $k \in \mathbb{N}_0$, i.e. the opening lies in $\mathbb{Z}[1/e]$ instead of $\mathbb{Q}$. In RSA groups, the hardness assumption is implied (with $e = 1$) by strong RSA, assuming safe primes are used, and therefore $x = m$, i.e. x is an *integer* representative. For class groups, the hardness assumption (with $e = 2$) is novel,[15] and see that $x = m/2^k$, i.e. a dyadic integer.

[15] More precisely, we require a family of assumptions, which collapses to two assumptions when one assumes invertible sampling. Moreover, we show the assumptions are closely related to the better understood 2-fROOT and ORD assumptions. (See lemma A.13) Unfortunately, our reductions do not apply for groups without invertible sampling. Thus, they cannot be used to provably justify security when using transparent setup. However, it is still a heuristic justification of their hardness.

We leverage this in our range proof by adding an additional commitment to x in the hidden order group $\mathbb{H}$, and proving consistency and small opening. For RSA groups, we get $x \in \mathbb{Z}$ and hence *standard soundness*. For class groups, we get $x = \frac{m}{2^k}$. Despite allowing a denominator, this is a huge improvement over arbitrary rational representative, as homomorphic computation now pose a much smaller threat, since the committed values are forced to lie in $\frac{1}{2^{\log(\Gamma)}}\mathbb{Z}$, which is closed under addition (unlike $\mathbb{Q}_{\infty,D}$ for general D).

C.2. Augmented Range Proof

The modification to our schemes is surprisingly lightweight. It reuses the challenges of the standard scheme and does not require repetitions. The only additional communication is the commitment to x and the masked randomness required for the proof of short opening.

As additional setup, a Pedersen commitment key $\mathrm{ck}_{\mathbb{H}} = (G'_i, \rho_i)_{i=0}^{N}$, where $(G'_i, \rho_i) \leftarrow \mathsf{Sample}(1^\lambda, \mathbb{H})$, with hiding parameter S is required. (We explicitly include the public coins ρ_i in the commitment key due to the lack of invertible sampling in class groups, cf. appendix A.3.1. Note that the correspondence of G'_i and ρ_i must be checked by the parties (once).) We mask values in $[0, S(\Gamma+1)^R]$ with masking algorithm mask'_r, masking randomness distribution R'_r, masking overhead L'_r and abort probability $\mathsf{p_r}'$. As stated above, the main difference is, that an additional MPed commitment C'_x to all x_i using $\mathrm{ck}_{\mathbb{H}}$ is made (and sent by the prover), and knowledge of opening of C'_x is proven. We first describe the more complex case of $\mathsf{Sharp}_{\mathsf{GS}}$.

C.2.1. Necessary Modifications to $\mathsf{Sharp}_{\mathsf{GS}}$.

We describe only the modifications of algorithm 1 ($\mathsf{Sharp}_{\mathsf{GS}}$) below, using the same variable names as in $\mathsf{Sharp}_{\mathsf{GS}}$.

- The prover's first flow (e.g. after line 12) is changed as follows: Additionally commit the x_i in $\mathbb{H}$.
  1. $C'_x = r'_x G'_0 + \sum_{i=1}^{N} x_i G'_i$, where $r'_x \xleftarrow{\$} [0, S]$.

  Now, compute the first message of the proof of short opening in $\mathbb{H}$.
  2. Set $\widetilde{r}'_x \xleftarrow{\$} \mathsf{R}'_r$ and let $\widetilde{x}_{k,i}$ be as in $\mathsf{Sharp}_{\mathsf{GS}}$.
  3. Let $\widetilde{x}'_i = \sum_{k=1}^{R} (\Gamma+1)^{k-1} \widetilde{x}_{k,i}$
  4. Set $D'_x = \widetilde{r}'_x G'_0 + \sum_{i=1}^{N} \widetilde{x}'_i G'_i$

  Modify the sent message as follows:
  1. Add C'_x to the message.
  2. With the hash optimization, add D'_x to the list of hashed messages. (Without it, add D'_x to the message.)

- The verifier's challenge is unmodified. Recall that $\gamma_k \in [0, \Gamma]$ for $k \in [1, R]$ are the challenges.

- The provers's response (e.g. after line 18) is changed as follows: Compute the "synthesized challenge" γ'.
  1. Set $\gamma' = \sum_{k=1}^{R} \gamma_k (\Gamma+1)^{k-1} \in [0, (\Gamma+1)^R - 1]$.

  Compute the masked opening randomness.
  2. Set $t'_x = \mathsf{mask}'_r(\gamma' \cdot r'_x, \widetilde{r}'_x)$.

  Abort if any masking failed. Modify the sent message as follows: Add t' to the message.

- The verifier's check (e.g. after line 8) is changed as follows. Let the "synthesized" challenge and responses be:
  1. $\gamma' = \sum_{k=1}^{R} \gamma_k (\Gamma+1)^{k-1} \in [0, (\Gamma+1)^R - 1]$
  2. $z'_i = \sum_{k=1}^{R} (\Gamma+1)^{k-1} \cdot z_{k,i}$

  Add the following computations and checks:
  3. Compute the "synthesized" γ' and z'_i for $i \in [1, N]$.
  4. Compute $F'_x = -\gamma' C'_x + t'_x \cdot G'_0 + \sum_{i=1}^{N} z'_i \cdot G'_i$
  5. With the hash optimization, modify the check of Δ by including F'_x in the list of messages. (Without it, check $F'_x = D_x$.)

C.2.2. Necessary modifications to $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$.

For $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$, the analogous changes of $\mathsf{Sharp}_{\mathsf{GS}}$ are applied to Phase 2. Since there are no repetitions in Phase 2, no "synthesized" γ' needed, hence $\gamma' = \gamma$ and $[0, \widehat{\Gamma}]$ is used as challenge space. We stress, that maskings which are shared with $\mathbb{H}$ (concretely, $z_i = \mathsf{mask}_x(\gamma x_i, \widetilde{x}_i)$) must now be computed over $\mathbb{Z}$ (and not modulo p, since the group orders of $\mathbb{H}$ and $\mathbb{G}_{\mathsf{com}}$ are "incompatible").

C.2.3. Efficiency.

We consider the schemes with hash optimization applied. Then, compared to $\mathsf{Sharp}_{\mathsf{GS}}$ (resp. $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$), the additional communication is a single element in $\mathbb{H}$ (namely C'_x), and the integer $t'_x \in \mathsf{R}'_r$. Additional computation for the prover's is computing C'_x and D'_x. For the verifier, it is the computation of F'_x. Other changes are negligible.

C.2.4. Security.

$\mathsf{Sharp}^{+\mathsf{HO}}_{\mathsf{GS}}$ is correct, non-abort SHVZK and provides a *strengthened* relaxed soundness guarantee. Informally, the committed x_i are guaranteed to have rational representatives in $[-\frac{1}{4B}, B + \frac{1}{4B}]\mathbb{Q}$, where due to hardness of (Γ, e, N)-relaxed DLOG-relations in $\mathbb{H}$, x_i is of the form m/e^ℓ for $m \in [-2(B\Gamma+1)L, 2(B\Gamma+1)L], \ell \leq \log(\Gamma)$. To deal with the lack of invertible sampling in the class group setting, we consider an explicit sampling algorithm Sample for uniform group elements, and hardness of assumptions w.r.t. Sample.

Theorem C.1. *Let* Sample *be a sampling algorithm for* $\mathbb{G}$*. The scheme* $\mathsf{Sharp}^{+\mathsf{HO}}_{\mathsf{GS}}$ *has correctness error at most* $1-(1-\mathsf{p_r}')^N[(1-\mathsf{p_r})^3 \cdot (1-\mathsf{p_x})^N]^R$*. It is non-abort SHVZK under the* SEI *assumptions on* $\mathbb{G}$ *and the* SEI *and the* SI *assumptions on* $\mathbb{H}$*. Is has relaxed soundness for the relation*

$$\begin{aligned}\mathsf{R}_{\mathsf{Ext}} = \{(x_1, \ldots, x_N, r) \colon\ & C_x = r_x G_0 + \sum_{i=1}^{N} x_i G_i \\ & \wedge\ \exists m_i \in \mathbb{Z}, k \in \mathbb{N}_0 \colon\ -\frac{1}{4B} \leq \frac{m_i}{e^k} \leq B + \frac{1}{4B} \\ & \wedge\ x_i \equiv_q \frac{m_i}{e^k} \wedge |m_i| \leq (B\Gamma+1)L_x \wedge 1 \leq e^k \leq \Gamma\}.\end{aligned}$$

under the DLOG, SEI *assumptions on* $\mathbb{G}$*, and the* DLOG, SEI, SI*, assumption and hardness of* (Γ, e)*-relaxed DLOG-relations in* $\mathbb{H}$*, where all asumptions are all w.r.t. to* Sample*. The knowledge error is* $(\frac{2}{\Gamma+1})^R$*. Concretely, with the hash-optimization, we have following reductions:*

- *For every adversary* $\mathscr{A}$ *against non-abort SHVZK, there are adversaries* $\mathscr{B}_{\mathbb{G},\mathsf{SEI}}$, $\mathscr{B}_{\mathbb{H},\mathsf{SEI}}$, $\mathscr{B}_{\mathbb{H},\mathsf{SI}}$ *whose run-time is roughly that of* $\mathscr{A}$ *and so that* $\mathsf{Adv}^{\text{na-hvzk}}_{\mathscr{A}} \leq \mathsf{Adv}^{\text{sei}}_{\mathbb{H},\mathscr{B}_{\mathbb{G}_{\mathsf{com}},\mathsf{SEI}}} + \mathsf{Adv}^{\text{sei}}_{\mathbb{H},\mathscr{B}_{\mathbb{H},\mathsf{SEI}}} + \mathsf{Adv}^{\text{si}}_{\mathbb{H},\mathscr{B}_{\mathbb{H},\mathsf{SI}}}$.

- *For every adversary $\mathscr{A}$ against knowledge which runs at most T steps, there are adversaries $\mathscr{B}_{CR}$, $\mathscr{B}_{\mathbb{G},\mathsf{DLOG}}$, $\mathscr{B}_{\mathbb{H},\mathsf{DLOG}}$, $\mathscr{B}_{\mathbb{H},\mathsf{SI}}$, $\mathscr{B}_{\mathbb{H},\mathsf{relDLOG}}$, whose* expected *run-time is roughly $3 \cdot R \cdot T$, and so that* $\mathsf{Adv}^{\mathrm{ke}}_{\mathscr{A}} \leq (\frac{2}{\Gamma+1})^R + \mathsf{Adv}^{\mathrm{crhf}}_{\mathsf{Hash},\mathscr{B}_{CR}} + \mathsf{Adv}^{\mathrm{dlog}}_{\mathbb{G},\mathscr{B}_{\mathbb{G},\mathsf{DLOG}}} + \mathsf{Adv}^{\mathrm{dlog}}_{\mathbb{H},\mathscr{B}_{\mathbb{H},\mathsf{DLOG}}} + \mathsf{Adv}^{\text{rel-dlog}}_{\mathbb{H},(\Gamma,e,N),\mathscr{B}_{\mathbb{H},\mathsf{relDLOG}}}$.

To be precise, we consider the S-bounded SEI *assumption in $\mathbb{G}$ and the S-bounded* SEI *assumption in $\mathbb{H}$.*

The analogous adaption of theorem E.1 holds for $\mathsf{Sharp}^{+\mathsf{HO}}_{\mathsf{PoSO}}$, where the same additional terms for reductions in $\mathbb{H}$ appear.

Correctness follows by inspection. The soundness follows essentially as for the unmodified $\mathsf{Sharp}_{\mathsf{GS}}$ (theorem E.1), except that hardness of (Γ, e)-relaxed DLOG-relations is used to additionally argue that $x_i \in \mathbb{Z}[1/e]$ as sketched in appendix C.1. Zero-knowledge follows almost exactly as for $\mathsf{Sharp}_{\mathsf{GS}}$. The full proof is in appendix E.3.

D. Proofs for Shortness Testing

D.1. Proof of lemma 3.6

Proof. Define $s_i \coloneqq \lceil \frac{ip}{d} \rceil$ for $i = 0, \ldots, d-1$. Observe that, by eq. (3.3),

$$s_i = \left\lceil \frac{ip}{d} \right\rceil = \frac{ip}{d} + \frac{ip \bmod d}{d}$$

and after multiplication by d,

$$ds_i = ip + (ip \bmod d).$$

This equality holds over $\mathbb{Z}$. Modulo p, we find $ds_i \equiv_p (ip \bmod d)$. Since $\gcd(d, p) = 1$, all $ip \bmod d$ are distinct, hence $[0, d-1] = \mathbb{Z}_d = \{ip \bmod d \mid i \in [0, d-1]\}$. Dividing by d (over $\mathbb{Z}_p$), we find that

$$\{s_0, \ldots, s_{d-1}\} \equiv_p \{j/d \in \mathbb{Z}_p \mid j \in [0, d-1]\}$$

Thus, we have shown that the set $\mathbb{S}_d$ indeed consists of the $s_i = \lceil \frac{ip}{d} \rceil$. The closest elements to s_i are $s_{(i+1 \bmod d)}$ or $s_{(i-1 \bmod d)}$, and (since the space is "circular") it suffices to consider the distances $s_{(i+1 \bmod d)} - s_{(i \bmod d)}$ to lower-bound the minimal distance δ in $\mathbb{S}_d$. For $i = 0, \ldots, d-2$, we find

$$\left\lfloor \frac{p}{d} \right\rfloor \leq \left\lceil \frac{(i+1)p}{d} \right\rceil - \left\lceil \frac{ip}{d} \right\rceil = s_{i+1} - s_i \leq \left\lceil \frac{p}{d} \right\rceil$$

as claimed. For $i = d-1$, we find $s_0 - s_{d-1} \equiv_p p - s_{d-1}$, and since $p = \left\lceil \frac{dp}{d} \right\rceil$, the claim follows as above. In fact, $p - \lceil \frac{(d-1)p}{d} \rceil = \lfloor \frac{p}{d} \rfloor$, since $\lceil \frac{(d-1)p}{d} \rceil + \lceil \frac{p}{d} \rceil = p + 1$ (since $d \nmid p$). □

D.2. Proof of lemma 3.7

Proof. First we show eq. (3.6). By lemma 3.6, the distance between points in $\mathbb{S}_d = \{\frac{i}{d} \bmod p \mid i \in [0, \ldots, d-1]\}$ is at least $\delta = \lfloor p/d \rfloor$. Consequently, at most $\left\lceil \frac{K+1}{\delta} \right\rceil$ points can lie in an interval with $K+1$ elements, e.g. $[0, K]_{\mathbb{Z}_p} + \mu$, by a simple counting argument.

The next claim, eq. (3.7) follows by a simple direct analysis. Namely, $u \bmod p$ is distributed almost uniformly over $[0, p-1]$, in particular $\rho(u/U_{\mathbb{Z}_p}) \leq 2$. Moreover, multiplication with $1/d$ is a bijection modulo p since $\gcd(d, p) = 1$, so $U_{\mathbb{Z}_p}/d \bmod p$ is distributed as $U_{\mathbb{Z}_p}$. Consequently, $\Pr[u/d \in_{\mathbb{Z}_p} [0, K]_{\mathbb{Z}_p}] \leq \rho(u/U_{\mathbb{Z}_p}) \cdot \Pr[U_{\mathbb{Z}_p} \in_{\mathbb{Z}_p} [0, K]_{\mathbb{Z}_p}] \leq 2\frac{K+1}{p}$.

Finally, eq. (3.8) follows by case distinction. Let $\widehat{K} \coloneqq K+1$. For $d > p$, it follows immediately from eq. (3.7). For $d < p/2$, we get

$$\left\lfloor \frac{\widehat{K}}{\delta} \right\rfloor = \left\lfloor \frac{\widehat{K}}{\lfloor \frac{p}{d} \rfloor} \right\rfloor \leq \frac{\widehat{K}}{\lfloor \frac{p}{d} \rfloor} \leq \frac{\widehat{K}}{p/d-1} \leq d \cdot \frac{\widehat{K}}{p-d} \leq d \cdot \frac{2(K+1)}{p}$$

since $p/d - 1 = (p-d)/d$ and $p - d \geq p/2$. Using $\lceil \widehat{K}/\delta \rceil \leq 1 + \lfloor \widehat{K}/\delta \rfloor$ it follows that $\frac{1}{d}\lceil \widehat{K}/\delta \rceil \leq \frac{1}{d}(1 + 2d\frac{K+1}{p})$. For $p/2 < d < p$, we have $\delta = \lfloor p/d \rfloor = 1$ and $1/d < 2/p$, hence $\frac{1}{d}\lceil \widehat{K}/\delta \rceil = \frac{\widehat{K}}{d} \leq 2\frac{K+1}{p}$. □

D.3. Proof of lemma 3.8

Proof. Consider

$$\begin{aligned}
&\Pr\big[u\frac{a}{b} \in_{\mathbb{Z}_p} [0,K]_{\mathbb{Z}_p} + \mu + \mathbb{S}_d\big] \\
\leq &\Pr\big[ua \in_{\mathbb{Z}_p} [0,bK]_{\mathbb{Z}_p} + \mu' + b \cdot \mathbb{S}_d\big] \\
\leq &\Pr\big[ua \in_{\mathbb{Z}_p} [0,bK]_{\mathbb{Z}_p} + \mu' + \mathbb{S}_{d/b} + [0,b-1]\big] \\
\leq &\Pr\big[ua \in_{\mathbb{Z}_p} [0,b(K+1)-1]_{\mathbb{Z}_p} + \mu' + \mathbb{S}_{d/b}\big]
\end{aligned}$$

where we used that $\mu' = b\mu$, $b \cdot [0,K] \subseteq [0,bK]$, and $b \cdot \mathbb{S}_d \subseteq \mathbb{S}_{d/b} + [0,b-1]$. (The latter follows since $b \cdot i/d = i/d'$ where $d' = d/b$, and $i < d$, so $i/d' = (i \bmod d')/d' + \lfloor i/d' \rfloor \in \mathbb{S}_{d/b} + [0,b-1]$.) For brevity, define $K' = b(K+1) - 1$.

Claim D.1. *If $u'a \in_{\mathbb{Z}_p} [0,K']_{\mathbb{Z}_p} + \mu' + s'$ for some choice $u' \in [0,D]$ and $s' \in \mathbb{S}_{d/b}$, then s' is unique, i.e. there exists no other choice $u'' \in [0,D]$, $s'' \in \mathbb{S}_{d/b}$ with $s' \neq s''$ and $u''a \in_{\mathbb{Z}_p} [0,K']_{\mathbb{Z}_p} + \mu' + s''$.*

Proof. Suppose otherwise. Observe that $ua \in [0,Da]$. Hence the distance of $u''a$ and $u'a$ is at most Da, Considering the "slack" of $[0,K']$, the points $s', s'' \in \mathbb{S}_{d/b}$ can therefore be as most $K' + Da$ far apart. The minimal distance in $\mathbb{S}_{d/b}$ is $\lfloor p/(d/b) \rfloor$. However, by assumption (eq. (3.9)) $K' + Da < \lfloor p/(d/b) \rfloor$. Thus, $s' \neq s''$ must be too far from each other, which is a contradiction. □

We have just shown that there is a at most one $s' \in \mathbb{S}_{d/b}$ for which $u'a \in_{\mathbb{Z}_p} [0,K']_{\mathbb{Z}_p} + \mu' + s'$ can happen. Thus, we find

$$\Pr\big[ua \in_{\mathbb{Z}_p} [0,K']_{\mathbb{Z}_p} + \mu' + \mathbb{S}_{d/b}\big] \leq \Pr\big[ua \in_{\mathbb{Z}_p} [0,K']_{\mathbb{Z}_p} + \mu''\big]$$

where $\mu'' = \mu' + s'$. But it is clear that at most $\lceil (K'+1)/a \rceil$ choices of u can lie in an the interval with $K'+1$ elements (since $Da < p$). With $K'+1 = b(K+1)$, it follows that

$$\Pr\big[ua \in_{\mathbb{Z}_p} [0,K']_{\mathbb{Z}_p} + \mu''\big] \leq \left\lceil \frac{b(K+1)}{a} \right\rceil \frac{1}{D+1}.$$

□

D.4. Proof of lemma 3.10

Lemma D.2. *Suppose $1 \neq d \in \mathbb{N}$ and let u_i be random variables in $\mathbb{Z}_d = [0,\ldots,d-1]$ for $i = 1,\ldots,N$. Fix some arbitrary $a_i \in [0,d-1]$ with $\operatorname{lcm}(a_1,\ldots,a_N) = d$. Define*

$$F\colon \mathbb{Z}_d \to \mathbb{Z}_d, \qquad F(u_1,\ldots,u_N) = \sum_{i=1}^{N} u_i \cdot a_i \bmod d \tag{D.1}$$

There exist $q_1,\ldots,q_N \in \mathbb{N}$ such that

1. *All q_i are coprime.*
2. $q_i \mid \mathsf{ord}_{\mathbb{Z}_d}(a_i)$.
3. $\prod_{i=1}^N q_i = d$.

Define $Z = \prod_{i=1}^N \mathbb{Z}_{q_i}$ and following homomorphisms:

- *The projections $\pi_i \colon \mathbb{Z}_d \to \mathbb{Z}_{q_i}$ and the CRT map $\pi \colon \mathbb{Z}_d \to Z$ with $\pi(x) = (\pi_1(x), \dots, \pi_N(x))$.*
- *The injections $\iota_i \colon \mathbb{Z}_{q_i} \to \mathbb{Z}_d$ defined by $x \mapsto \alpha_i \cdot x \bmod d$, where $\alpha_i \coloneqq \frac{d}{q_i} \cdot ((\frac{d}{q_i})^{-1} \bmod q_i) \in \mathbb{Z}$, and the combined injections $\iota \colon Z \to \mathbb{Z}_d^N$ as $\iota((x_1, \dots x_N)) = (\iota_1(x_1), \dots, \iota_N(x_N))$.*
- *The CRT map $\phi \colon Z \to \mathbb{Z}_d$, $\phi((x_1, \dots, x_N)) = \sum_{i=1}^N \iota_i(x_i) = \sum_{i=1}^N \alpha_i x_i$.*

Recall that π and ϕ are the bijections of the Chinese remainder theorem (CRT).
With this, we have:

4. *Restricted to $\iota(Z)$, the map $f \colon \iota(Z) \to \mathbb{Z}_d$, $f = F|_{\iota(Z)}$ is an isomorphism.*
5. *For uniform $(v_1, \dots, v_N) \xleftarrow{\$} Z$, we find that*

$$F(\iota(v_1, \dots, v_N)) = f(v_1, \dots, v_N) = \sum_{i=1}^N \iota_i(v_i) \cdot a_i \bmod d$$

is uniform in $\mathbb{Z}_d$. Consequently, for uniform $(u_1, \dots, u_N) \xleftarrow{\$} \iota(Z) \leq \mathbb{Z}_d^N$, also $F(u_1, \dots, u_N)$ is uniform in $\mathbb{Z}_d$.

The lemma is essentially an application of the Chinese remainder theorem (CRT) and some standard computations. We provide a small example: Suppose $N = 2$ and $d = 300 = 2^2 \cdot 3 \cdot 5^2$, $a_1 = 15$ and $a_2 = 4$. Then $\mathsf{ord}_{\mathbb{Z}_d}(a_1) = 300/15 = 2^2 \cdot 5$ and $\mathsf{ord}_{\mathbb{Z}_d}(a_2) = 300/20 = 3 \cdot 5^2$. Thus, let $q_1 = 2^2$ and $q_2 = 3 \cdot 5^2$ (i.e. gather the largest prime powers in the q_i's). Clearly, $\mathbb{Z}_{300} \cong \mathbb{Z}_4 \times \mathbb{Z}_{75}$ by the CRT. It's easy to check the claims as well; they are almost directly implied by the CRT.

Proof. First, we show the existence of q_i's as claimed and some resulting properties. For this, let $h_i = \mathsf{ord}_{\mathbb{Z}_d}(a_i)$. Let $h_i = p_1^{e_{i,1}} \cdot \dots p_r^{e_{i,r}}$ for distinct primes p_j and exponents $e_{i,j} \in \mathbb{N}_0$. Define q_1 as the product of those $p_j^{e_{1,j}}$ where $e_{1,j}$ is the maximal exponent (over all $j = 1, \dots r$). The other q_i are defined analogously. If for fixed fixed j, there are multiple i such that h_i has the *maximal* exponent $e_{i,j}$ for p_j, then $p_j^{e_{i,j}}$ is part of (only!) the q_i with the smallest index i. By construction, $q_1, \dots, q_N$ satisfy the required properties.

Conversely, the required properties enforce this structure, up to choices where for fixed j, multiple indices i have the maximal prime power $p_i^{e_{i,j}}$. This essentially follows from q_is being coprime, hence each prime (power) appears in at most one q_i, and $\prod_i^d q_i = d$, hence each prime (power) appears in at least one q_i.

Lastly, note that from the abstract properties, we get $(a_i \bmod q_i) \in \mathbb{Z}_{q_i}^\times$. This can be seen via the CRT: We have $\mathbb{Z}_d = \mathbb{Z}_{d/q_i} \times \mathbb{Z}_{q_i}$ via the CRT, since $\gcd(q_i, d/q_i) = 1$ by definition of q_i. Moreover, projecting a_i to $\mathbb{Z}_{q_i}$, a_i is must be generator of $\mathbb{Z}_{q_i}$ (or $q_i \nmid \mathsf{ord}_{\mathbb{Z}_d}(a_i)$, a contradiction).

Now we turn to Item 4. By the CRT, we have $Z = \prod_{i=1}^N \mathbb{Z}_{q_i} \cong \mathbb{Z}_{\prod_i q_i} = \mathbb{Z}_d$, and the isomorphism connecting Z and $\mathbb{Z}_d$ are π and ϕ. Moreover,

$$F(\iota(v_1, \dots, v_N)) = \sum_{i=1}^N \iota_i(v_i) \cdot a_i = \sum_{i=1}^N \alpha_i v_i \cdot a_i = \sum_{i=1}^N v_i \cdot \frac{d}{q_i} a_i'$$

where $a_i' = a_i((\frac{d}{q_i})^{-1} \bmod q_i)$ by definition of α_i. The order of $\frac{d}{q_i} a_i' = \alpha_i a_i$ in $\mathbb{Z}_d$ is exactly q_i (since α_i is invertible modulo q_i). As $\mathbb{Z}_d$ is cyclic, each subgroup is uniquely identified by its

order, and we conclude that the image $F(\iota(0,\dots,\mathbb{Z}_{q_i},0,\dots))$ is *the* subgroup of order q_i in $\mathbb{Z}_d$. Since $\prod_{i=1}^N q_i = d$, these subgroups span $\mathbb{Z}_d$ (again, by the CRT) and therefore f is surjective (and hence, bijective since $|Z| = |\mathbb{Z}_d|$).

Lastly, item 5 follows immediately from $f\colon Z \to \mathbb{Z}_d$ being an isomorphism, so in particular a bijection. □

D.5. Proof of lemma 3.11

Proof. We assume w.l.o.g. that $D \geq 2$, $K \geq 1$, $d > 1$ and $N > 1$; the excluded cases are straightforward. As a first step, we impose conditions on I, N and d.

Claim D.3. *We can w.l.o.g. assume that $I = \{1,\dots,N\}$ and that for any subset I' of I, $\operatorname{lcm}(d_{i'} \mid i' \in I) < \operatorname{lcm}(d_i \mid i \in I)$, that is, I is a minimal subset w.r.t. the least common multiple $d = \operatorname{lcm}(d_i \mid i \in I)$.*

Proof. First of all, we show that w.l.o.g. $I = \{1,\dots,N\}$. For this, note that

$$\sum_{i=1}^{N} \gamma_i \frac{m_i}{d_i} \in_{\mathbb{Z}_p} [0,K]_{\mathbb{Z}_p} + \mu \iff \sum_{i\in I} \gamma_i \frac{m_i}{d_i} \in_{\mathbb{Z}_p} [0,K]_{\mathbb{Z}_p} + \underbrace{(\mu - \sum_{i\notin I} \gamma_i \frac{m_i}{d_i})}_{\mu'}.$$

Thus, if the lemma holds only for the "partial-sum" $I \subseteq \{1,\dots,N\}$, it follows for the "complete sum" over $\{1,\dots,N\}$, by conditional probability and using that the bounds in eq. (3.12) must hold for *arbitrary* μ, in particular μ' (by induction over the instance size N). Thus, w.l.o.g. $I = \{1,\dots,N\}$.

Now, suppose removing some d_i, w.l.o.g. d_N, does not affect the least common multiple, i.e. $\operatorname{lcm}(d_{i'} \mid i' \in I \setminus \{N\}) = d$. Then instead of I we could use $I' = I \setminus \{N\}$. By the above, w.l.o.g. we can assume $I' = \{1,\dots,N\}$ again. Overall, we can assume "minimality" of I and $N = |I|$. □

Since w.l.o.g. $I = \{1,\dots,N\}$, from now on we will mostly ignore the index set I.

Before diving into the proof, recall that

$$\frac{a}{b} = \frac{a \bmod b}{b} + \left\lfloor \frac{a}{b} \right\rfloor. \tag{D.2}$$

Let $d = \operatorname{lcm}(\{d_i \mid i \in I\})$ as in the claim. To motivate our approach, we first rewrite eq. (3.11) with common denominator d and apply eq. (D.2) to find

$$\begin{aligned}
&\Pr[S \in [0,K]_{\mathbb{Z}_p} + \mu] \\
&= \Pr\left[\sum_{i=1}^{N} \gamma_i \cdot \frac{m_i}{d_i} \in_{\mathbb{Z}_p} [0,K]_{\mathbb{Z}_p} + \mu\right] \\
&= \Pr\left[\frac{1}{d}\left(\sum_{i=1}^{N} \gamma_i \cdot m_i \frac{d}{d_i}\right) \in_{\mathbb{Z}_p} [0,K]_{\mathbb{Z}_p} + \mu\right] \\
&= \Pr\left[\frac{1}{d}\left(\left(\sum_{i=1}^{N} \gamma_i \cdot m_i \frac{d}{d_i}\right) \bmod d\right) + \left\lfloor \sum_{i=1}^{N} \gamma_i m_i \frac{1}{d_i} \right\rfloor \in_{\mathbb{Z}_p} [0,K]_{\mathbb{Z}_p} + \mu\right]
\end{aligned}$$

Observe that we now have a sum modulo d and are *almost* in the situation of lemma D.2, which in turn would allow us to apply lemma 3.7. But $\sum_{i=1}^N \gamma_i \cdot m_i \frac{d}{d_i} \bmod d$ need not be uniform modulo d, and $\lfloor \sum_{i=1}^N \gamma_i m_i \frac{1}{d_i} \rfloor$ is a stochastically dependent "error term". Thus, we will change the distribution of the γ_i in a suitable manner to obtain two independent sums.

For better tightness, we use the distribution suggested by lemma D.2. Let q_i be as in lemma D.2. Together with claim D.3, we get the following properties.

Claim D.4. *We have that all q_i are coprime, $q_i > 1$ for all i, $\prod_{i=1}^{N} q_i = d$, $q_i \mid \mathsf{ord}_{\mathbb{Z}_d}(m_i d/d_i) \mid d_i \mid d$, and $N \leq \mathsf{primlmin}(D+1)$.*

Proof. Directly from lemma D.2, we see that all q_i are coprime, $q_i \mid \mathsf{ord}_{\mathbb{Z}_d}(m_i d/d_i)$, and $\prod_{i=1}^{N} q_i = d$. The divisibility chain is completed via $\mathsf{ord}_{\mathbb{Z}_d}(m_i d/d_i) \mid \mathsf{ord}_{\mathbb{Z}_d}(d/d_i) = d_i \mid d$. To see $q_i > 1$, suppose to the contrary that some $q_j = 1$. Then $\prod_{i \neq j} q_i = d = \mathrm{lcm}(\{d_i\}_{i \neq j})$. But this contradicts the minimality of the index set I (and N) which we established w.l.o.g. in claim D.3.

To see $N \leq \mathsf{primlmin}(D+1)$, observe that each q_i contributes different prime factors, and therefore $\mathsf{priml}(k) = \prod_{i=1}^{k} p_i \leq \prod_{i=1}^{k} q_i = d$, where p_i denotes the i-th prime number. Hence, if $\mathsf{priml}(k) \geq D+1$, then $d > D$, and therefore $k \leq \mathsf{primlmin}(D+1)$. □

Claim D.4 explains why $\beta = \min(|I|, \mathsf{primlmin}(D+1))$ is used in lemma 3.11, because in lemma 3.11 no assumptions on "minimality" of I (and N) were made (in particular $N > \mathsf{primlmin}(D+1)$ is possible).

Now, we change the distribution which we consider from $\gamma_i \xleftarrow{\$} [0, D]$ to $\gamma_i' \xleftarrow{\$} [0, q_i\lceil (D+1)/q_i \rceil - 1]$. (Observe that $q_i\lceil (D+1)/q_i \rceil \geq D+1$ is the smallest multiple of q_i which is larger or equal to $D+1$, and $\gamma_i' \bmod q_i$ is uniformly distributed.)

To simplify notation, let $\widehat{D} \coloneqq D+1$, i.e. $\widehat{D}$ is the cardinality of $[0, D]$. One quickly computes

$$\rho(\gamma_i/\gamma_i') = \frac{1}{\widehat{D}} \cdot \left(\frac{1}{q_i \lceil \widehat{D}/q_i \rceil} \right)^{-1} = \frac{q_i \lceil \widehat{D}/q_i \rceil}{\widehat{D}} \leq 1 + \frac{q_i - 1}{\widehat{D}}$$

where we use that

$$0 \leq q_i \lceil \widehat{D}/q_i \rceil - \widehat{D} = (\widehat{D} \bmod q_i) \leq q_i - 1.$$

Observe that we can sample and write $\gamma_i' \xleftarrow{\$} [0, q_i\lceil (D+1)/q_i \rceil - 1]$ as

$$\gamma' = u_i + q_i v_i \quad \text{where} \quad u_i \xleftarrow{\$} [0, q_i - 1], \quad v_i \xleftarrow{\$} [0, \lceil \widehat{D}/q_i \rceil - 1].$$

For future reference, we record the following definitions and facts.

Claim D.5. *Let $u_i \xleftarrow{\$} [0, q_i - 1]$, $v_i \xleftarrow{\$} [0, \lceil \widehat{D}/q_i \rceil - 1]$ with q_i as above. Define*

$$S_u \coloneqq \sum_{i=1}^{N} u_i m_i \frac{d}{d_i} \quad \text{and} \quad S_v \coloneqq \sum_{i=1}^{N} v_i m_i \frac{q_i}{d_i} \tag{D.3}$$

Then $S_u \bmod d$ is uniform in $\mathbb{Z}_d$.

Proof. The claim is immediate by lemma D.2 (and definition of q_i, u_i). □

Now, let

$$\rho \coloneqq \rho((\gamma_1, \ldots, \gamma_N)/(\gamma_1', \ldots, \gamma_N')) \leq \prod_{i=1}^{N} (1 + \frac{q_i - 1}{\widehat{D}}) \tag{D.4}$$

Claim D.6. *It holds that $\rho \leq 4$.*

Proof. As claim D.6 follows from unrelated technical computations, we prove this separately in lemma D.10, which only needs the following constraints, already observed in (the proof of) claim D.4,

- $1 < q_i \leq D$ for all $i = 1, \ldots, N$.
- $\prod_{i=1}^{N} q_i < D^2$ and for all subset products over $I' \subsetneq \{1, \ldots, N\}$ it holds that $\prod_{i \in I'} q_i \leq D$.

□

Now that notation and setup are in place, we turn to the following central claim.

Claim D.7. *It holds that*

$$\Pr[S \in [0,K]_{\mathbb{Z}_p} + \mu] \leq \rho \cdot \Pr\left[\frac{1}{d}(S_u \bmod d) + S_v \in_{\mathbb{Z}_p} [0,K']_{\mathbb{Z}_p} + \mu'\right]$$

where

$$K' = K + 2\beta M \quad \textit{and} \quad \mu' = \mu - \beta M, \tag{D.5}$$

for $\beta = \min(N, \mathsf{primlmin}(D+1))$.

Proof. From our definition of γ'_i and eq. (D.4), we get

$$\begin{aligned}
&\Pr[S \in [0,K]_{\mathbb{Z}_p} + \mu] \\
&= \Pr\left[\sum_{i=1}^{N} \gamma_i \cdot \frac{m_i}{d_i} \in_{\mathbb{Z}_p} [0,K]_{\mathbb{Z}_p} + \mu\right] \\
&\leq \rho \cdot \Pr\left[\sum_{i=1}^{N} \gamma'_i \cdot \frac{m_i}{d_i} \in_{\mathbb{Z}_p} [0,K]_{\mathbb{Z}_p} + \mu\right] \\
&= \rho \cdot \Pr\left[\sum_{i=1}^{N} (u_i + q_i v_i) \cdot \frac{m_i}{d_i} \in_{\mathbb{Z}_p} [0,K]_{\mathbb{Z}_p} + \mu\right] \\
&= \rho \cdot \Pr\left[\frac{1}{d} S_u + S_v \in_{\mathbb{Z}_p} [0,K]_{\mathbb{Z}_p} + \mu\right]
\end{aligned} \tag{D.6}$$

where we first use the properties of $\rho(\cdot/\cdot)$ to replace γ_i by γ'_i, then we use $\gamma'_i = u_i + q_i v_i$ and rewrite the sum. As usual (by eq. (D.2)), we have

$$\begin{aligned}
\frac{1}{d} S_u &= \frac{1}{d}(S_u \bmod d) + \left\lfloor \frac{S_u}{d} \right\rfloor \\
&= \frac{1}{d}\left(\sum_{i=1}^{N} u_i \cdot m_i \frac{d}{d_i} \bmod d\right) + \left\lfloor \sum_{i=1}^{N} u_i \cdot m_i \frac{1}{d_i} \right\rfloor.
\end{aligned}$$

We first derive a bound for $\left\lfloor \frac{S_u}{d} \right\rfloor$. Note that

$$\left\lfloor \sum_{i=1}^{N} u_i \cdot m_i \frac{1}{d_i} \right\rfloor \leq \left\lfloor \sum_{i=1}^{N} m_i \frac{q_i - 1}{d_i} \right\rfloor \leq \left\lfloor M \cdot \sum_{i=1}^{N} \frac{q_i - 1}{d_i} \right\rfloor \leq \left\lceil \sum_{i=1}^{N} \frac{q_i - 1}{d_i} \right\rceil \cdot M$$

where $\alpha = \lceil \sum_{i=1}^{N} \frac{q_i - 1}{d_i} \rceil \leq N$, since $(q_i - 1)/d_i < 1$ by choice of q_i (namely, $q_i \mid d_i$). Analogously,

$$\left\lfloor \sum_{i=1}^{N} u_i \cdot m_i \frac{1}{d_i} \right\rfloor \geq \left\lfloor -M \cdot \sum_{i=1}^{N} \frac{q_i - 1}{d_i} \right\rfloor \geq -\left\lceil \sum_{i=1}^{N} \frac{q_i - 1}{d_i} \right\rceil \cdot M$$

hence $-\alpha M$ is a lower bound. Thus, we find

$$\left\lfloor \frac{S_u}{d} \right\rfloor = \left\lfloor \sum_{i=1}^{N} u_i \cdot m_i \frac{1}{d_i} \right\rfloor \in [-\alpha M, \alpha M]. \tag{D.7}$$

That is, the possible values of the sum lie in the interval $[-\alpha M, \alpha M] = [0, 2\alpha M] - \alpha M$. Note that, by claim D.4 and our simplifying assumptions on N and I in claim D.3, $\alpha \leq \min(N, \mathsf{primlmin}(D+1)) = \beta$ holds.

Now, we can continue eq. (D.6) with

$$\begin{aligned}
&\rho \cdot \Pr\left[\frac{1}{d} S_u + S_v \in_{\mathbb{Z}_p} [0, K]_{\mathbb{Z}_p} + \mu\right] \\
&= \rho \cdot \Pr\left[\frac{1}{d}(S_u \bmod d) + \left\lfloor \frac{S_u}{d} \right\rfloor + S_v \in_{\mathbb{Z}_p} [0, K]_{\mathbb{Z}_p} + \mu\right] \\
&\leq \rho \cdot \Pr\left[\frac{1}{d}(S_u \bmod d) + S_v \in_{\mathbb{Z}_p} [0, K']_{\mathbb{Z}_p} + \mu'\right]
\end{aligned}$$

where we first used eq. (D.2) as usual, and then eq. (D.7), as well as the definition $K' = K + 2\beta M$ and $\mu' = \mu - \beta M$. This proves claim D.7. □

We are now in a position to prove lemma 3.11. We first show eq. (3.12) of lemma 3.11.

Claim D.8. *It holds that*

$$\Pr[S \in [0, K]_{\mathbb{Z}_p} + \mu] \leq \rho \cdot \begin{cases} \frac{1}{d} & \text{if } d(K'+1) < p \\ \frac{1}{d} + 2\frac{K'+1}{p} & \text{always} \end{cases}$$

From claim D.8 the first claim of the core lemma, eq. (3.12), follows using $\rho \leq 4$ from claim D.6.

Proof. Using claim D.7, it suffices to prove that

$$\varepsilon \coloneqq \Pr\left[\frac{1}{d}(S_u \bmod d) + S_v \in_{\mathbb{Z}_p} [0, K']_{\mathbb{Z}_p} + \mu'\right] \leq \frac{1}{d} + 2\frac{K'+1}{p}$$

and $\varepsilon \leq 1/d$ if $d(K'+1) < p$.

Since by construction, u_i and v_i are stochastically independent, we find

$$\varepsilon \leq \sum_{t \in \mathbb{Z}_p} \Pr\left[\frac{1}{d}(S_u \bmod d) \in_{\mathbb{Z}_p} [0, K']_{\mathbb{Z}_p} + \mu' - t\right] \cdot \Pr[S_v \equiv_p t].$$

Now, recall that $S_u \bmod d$ is uniformly distributed in $\mathbb{Z}_d$ (cf. claim D.5), indeed, this was the reason for switching from γ_i to γ_i'. Thus,

$$\Pr\left[\frac{1}{d} S_u \in_{\mathbb{Z}_p} [0, K']_{\mathbb{Z}_p} + z'\right] = \Pr\left[\frac{1}{d} \cdot U_{\mathbb{Z}_d} \in_{\mathbb{Z}_p} [0, K']_{\mathbb{Z}_p} + z'\right] \leq \frac{1}{d} + 2\frac{K'+1}{p}$$

where $z' = \mu' - t$ and the inequality follows from lemma 3.7. Hence,

$$\varepsilon \leq \sum_{t \in \mathbb{Z}_p} \frac{1}{d} \cdot \Pr[S_v \equiv_p t] = \frac{1}{d} + 2\frac{K'+1}{p}$$

and this part of the claim follows. Moreover, by eq. (3.6) of lemma 3.7, if $\lceil (K'+1)/\lfloor p/d \rfloor \rceil \leq 1$, then $\varepsilon \leq 1/d$ (by the same argument). Since $\lfloor x \rfloor \leq x$, we can simplify to $\lceil (K'+1)/\lfloor p/d \rfloor \rceil \leq \lceil d(K'+1)/p \rceil \leq 1$, and thus, $d(K'+1)/p \leq 1$, as claimed. □

Finally, we turn to proving eq. (3.13) of lemma 3.11.

Claim D.9. *Suppose $d \leq D$ and there is some i^* such that $dm_{i^*}/d_{i^*} > K' = K + 2\beta M$. Then*

$$\Pr[S \in_{\mathbb{Z}_p} [0, K]_{\mathbb{Z}_p} + \mu] \leq \frac{8}{D+1}.$$

Proof. Again, we continue from claim D.7 and use independence of the u_is and v_is to find

$$\begin{aligned}
&\Pr\left[\frac{1}{d}(S_u \bmod d) + S_v \in_{\mathbb{Z}_p} [0, K']_{\mathbb{Z}_p} + \mu'\right] \\
&= \sum_{t \in \mathbb{S}_d} \Pr\left[\frac{1}{d}(S_u \bmod d) = t\right] \cdot \Pr\left[S_v \in_{\mathbb{Z}_p} [0, K']_{\mathbb{Z}_p} + \mu' - t\right] \\
&= \frac{1}{d} \sum_{t \in \mathbb{S}_d} \Pr\left[S_v \in_{\mathbb{Z}_p} [0, K']_{\mathbb{Z}_p} + \mu' - t\right]
\end{aligned}$$

where $\mathbb{S}_d = \{i/d \bmod p \mid i \in [0, d-1]\} \subseteq \mathbb{Z}_p$. In the last equality, we use again that $S_u \bmod d$ is uniform in $\mathbb{Z}_d$ by claim D.5. Using the independence of the v_i's we further condition on all but i^*, and find

$$\begin{aligned}
&\frac{1}{d} \sum_{t \in \mathbb{S}_d} \Pr\left[S_v \in [0, K']_{\mathbb{Z}_p} + \mu' - t\right] \\
&= \frac{1}{d} \sum_{t \in \mathbb{S}_d} \Pr\left[\sum_{i=1}^{N} v_i \cdot m_i \frac{q_i}{d_i} \in [0, K']_{\mathbb{Z}_p} + \mu' - t\right] \\
&= \sum_{y \in \mathbb{Z}_p} \Bigg(\Pr\left[\sum_{i \neq i^*} v_i \cdot m_i \frac{q_i}{d_i} = y\right] \cdot \\
&\qquad \frac{1}{d} \sum_{t \in \mathbb{S}_d} \underbrace{\Pr\left[v_{i^*} \cdot m_{i^*} \frac{q_{i^*}}{d_{i^*}} \in [0, K'] + \mu' - t - y\right]}_{(\star)} \Bigg)
\end{aligned} \tag{D.8}$$

To improve readability, we abbreviate terms with index i^* as v^*, m^*, etc. Now, we want to apply lemma 3.8 to $(\star)$, where $a \mathrel{\widehat{=}} q^* \cdot |m^*|$ and $b \mathrel{\widehat{=}} d^*$, and $D \mathrel{\widehat{=}} \lceil (D+1)/q^* \rceil - 1$, and $K \mathrel{\widehat{=}} K'$ and $\mu \mathrel{\widehat{=}} \mu' - y$. First, observe that the requirement

$$(K+1) + D\frac{a}{b} < \frac{1}{b} \cdot \left\lfloor \frac{p}{d/b} \right\rfloor$$

of lemma 3.8 is satisfied when the corresponding variables are inserted (by our premise on K, N, D, p). Namely, since $\frac{1}{b} \cdot \lfloor \frac{p}{d/b} \rfloor \leq \frac{p}{d} - \frac{1}{b}$ and $\frac{1}{b} \leq 1$, it suffices to see $2 + K + D\frac{a}{b} \leq \frac{p}{d}$. With $d \leq D$, $\frac{a}{b} \leq M$ we arrive at $D(K' + DM + 2) < p$, which holds by assumption. Thus, by the conclusion of lemma 3.8, we find

$$\begin{aligned}
&\frac{1}{d} \cdot \sum_{t \in \mathbb{S}_d} \Pr\left[v^* \cdot m^* \frac{q^*}{d^*} \in [0, K'] + \mu' - t - y\right] \\
\leq &\frac{1}{d} \cdot \left\lceil \frac{d^*(K'+1)}{m^* q^*} \right\rceil \frac{1}{\lceil (D+1)/q^* \rceil}.
\end{aligned}$$

Since, by assumption $dm^*/d^* < K'$, one can check that[16]

$$\frac{1}{d} \left\lceil \frac{d^*(K'+1)}{m^* q^*} \right\rceil \frac{1}{\lceil (D+1)/q^* \rceil} \leq \frac{\frac{q^*}{d} + \frac{d^*(K'+1)}{dm^*}}{D+1} \leq \frac{2}{D+1}.$$

[16] Let $A = \frac{1}{d} \lceil \frac{d^*(K'+1)}{m^* q^*} \rceil$. Let $B = 1/\lceil (D+1)/q^* \rceil$. We have to show $A/B \leq 2/(D+1)$. First, note that $B \leq q^*/(D+1)$ since $1/\lceil x \rceil \leq 1/x$. Using $\lceil x \rceil \leq x + 1$ in A, we find $A/B \leq \frac{1}{D+1}(\frac{d^*(K'+1)}{dm^*} + \frac{q^*}{d})$. Using $q^* \mid d^* \mid d$ and $d^* \neq d$ (since $N > 1$), we know $\frac{q^*}{d} \leq \frac{d^*}{d} \leq \frac{1}{2}$. Moreover, since $dm^*/d^* < K'$, i.e. $dm^*/d^* \leq K' + 1$, holds by assumption, we find $dm^* \leq d^*(K'+1)$ and hence $\frac{d^*(K'+1)}{dm^*} \leq 1$. All in all, $A/B \leq (1 + \frac{1}{2})/(D+1) \leq 2/(D+1)$.

Plugging that bound back into eq. (D.8), we obtain

$$\frac{1}{d}\sum_{t\in\mathbb{S}_d}\Pr[S_v \in [0,K']_{\mathbb{Z}_p} + \mu' - t] \leq \frac{2}{D+1}$$

and the claim follows since

$$\Pr[S \in [\mu,\mu+K]_{\mathbb{Z}_p}] \leq \rho \cdot \frac{1}{d}\sum_{t\in\mathbb{S}_d}\Pr[S_v = t] \leq \frac{8}{D+1}.$$

□

This finishes the proof of claim D.9, and hence of the lemma 3.11. □

Lemma D.10. *Let $D, N \in \mathbb{N}$ and $q_i \in \mathbb{N}$ with $2 \leq q_i \leq D$ for $i = 1,\ldots,N$. Suppose that $\prod_{i=1}^N q_i < D^2$, that $q_1 \geq \ldots \geq q_N$, and that any subset product is at most D. Then*

$$\prod_{i=1}^{N}(1+\frac{q_i-1}{D+1}) \leq \prod_{i=1}^{N}(1+\frac{q_i}{D}) \leq 4$$

Unfortunately, we cannot provide much intuition for lemma D.10 besides a proof overview: Namely, either $N = 2$, in which case the claim holds since $1 + q_i/D \leq 2$. Or $N > 2$. In that case, $\prod_{i=2}^N q_i \leq D$ and $q_2 \leq \sqrt{D}$ (since $q_1 \geq q_2$), by the "subset product premise". From this, it is easy to show that $\prod_{i=2}^N(1+\frac{q_i}{D}) \leq 2$ holds for "big enough" D (namely $D > 4$). The remaining cases (namely $D \leq 4$) are checked exhaustively.

Proof. We start with a simpler claim.

Claim D.11. *Let $a \geq b \geq 1$ and $\tau = ab$. Let $a' \geq b' \geq 1$ with $a' > a$ and $a'b' = \tau$. Then*

$$(1+a/D)(1+b/D) \leq (1+a'/D)(1+b'/D). \tag{D.9}$$

Proof. This follows by multiplying out both sides, subtracting the common term $1 + \tau/D^2$ on both sides, and multiplying with D to obtain the equivalent condition $a + b \leq a' + b'$. Using $ab = a'b' = \tau$, this becomes

$$a + \tau/a \leq a' + \tau/a'$$

and for $f(x) = x + \tau/x$ it is readily seen that f is monotonely increasing on domain $[\sqrt{\tau},\infty)$. Thus, the claim follows from $a' > a$ and $a \geq \sqrt{\tau}$ (which holds since $ab = \tau$ and $a \geq b$). □

The claim extends to products $\prod_{i=1}^N(1+\frac{a_i}{D})$ in the following manner: Consider $(a_1,\ldots,a_N)$ with $a_1 \geq \ldots \geq a_N \geq 1$ and $\prod_{i=1}^N a_i = \tau$ and $(a'_1,\ldots,a'_N)$ with analogous constraints. Suppose that $(a_i)_i$ and $(a'_i)_i$ differ only in components j_1 and j_2 with $j_1 < j_2$, and that $a_{j_1} < a'_{j_1}$. Then $\prod_{i=1}^N(1+a_i/D) < \prod_{i=1}^N(1+\frac{a'_i}{D})$ by claim D.11.

As a simple consequence, to maximize a product of the form $\prod_{i=1}^N(1+a_i/D)$ with constraints $a_i \in [2,M]_{\mathbb{R}}$ and $\prod_{i=1}^N a_i = \tau$, one must use a (permutation of) $(M,\ldots,\tau/(2^{N-\ell-1}M^\ell),2,\ldots,2)$, where ℓ is maximal.

Now, we return to prove the lemma. Let $\tau = \prod_{i=1}^N q_i$ and note that the product $\prod_{i=1}^N(1+\frac{q_i}{D})$ is maximized by maximizing q_1 and q_2 (due to $q_i \leq D$ and $q_1q_2 \leq \tau < D^2$).

It is useful to deal with following special case first:

Claim D.12. *Suppose that $D \geq 5$, $q_1 \leq \sqrt{D}$ and $\prod_{i=1}^N q_i \leq D$. Then $\prod_{i=1}^N(1+\frac{q_i}{D}) \leq 2$.*

Proof. The claim evidently holds for $N = 1$. It also holds for $N = 2$. Indeed, for $N = 2$ and any (fixed) $q_1 \geq q_2$, setting $D = q_1 q_2$ is the worst case. For this, one obtains $(1+\frac{q_1}{D})(1+\frac{q_2}{D}) = \frac{q_1+1}{q_1}\frac{q_2+1}{q_2}$, and for $D \geq 5$, this is at most 2.[17] The claim also holds for arbitrary N if $5 \leq D \leq 15$.[18] Thus, suppose $D \geq 16$ and $N \geq 3$. By the discussion after claim D.11, we find

$$\begin{aligned}\prod_{i=1}^{N}(1+\frac{q_i}{D}) &\leq \left(1+\frac{\sqrt{D}}{D}\right)\cdot\left(1+\frac{\sqrt{D}/2^{N-2}}{D}\right)\cdot\left(1+\frac{2}{D}\right)^{N-2} \\ &\leq \left(1+\frac{3/2\sqrt{D}+1}{D}\right)\cdot\left(1+\frac{2}{D}\right)^{N-2}\end{aligned}$$

where we maximized q_1 and q_2 (over $\mathbb{R}$) under the constraints that $q_i \geq 2$ and $\prod_{i=1}^{N} q_i \leq D$ and $q_1 \leq \sqrt{D}$ for all i. From $(1 + x/k)^k \leq e^x$ for $x \geq 0$, $k \in \mathbb{N}$, we find

$$\left(1+\frac{2}{D}\right)^{N-2} = \left(\left(1+\frac{2}{D}\right)^{D}\right)^{(N-2)/D} \leq e^{2(N-2)/D}.$$

From, $D \geq \prod_{i=1}^{N} q_i \geq 2^N$, we get $N \leq \log(D)$, and thus $2(N-2)/D \leq 2(\log(D)-2)/D$. Moreover, $f(x) = 2(\log(x)-2)/x$ is maximized at $x = 4e \leq 11$, with $f(4e) \lessapprox 0.2654$ we find

$$e^{2(N-2)/D} \leq e^{2(\log(D)-2)/D} \leq e^{0.2654} \leq 4/3.$$

Furthermore $(1+\frac{3/2\sqrt{D}+1}{D})$ is monotonely decreasing, hence $(1+\frac{3/2\sqrt{D}+1}{D}) \leq 1+\frac{7}{16}$ for $D \geq 16$. Thus, for $D \geq 16$, we find

$$\prod_{i=1}^{N}(1+\frac{q_i}{D}) \leq \left(1+\frac{7}{16}\right)\cdot 4/3 < 2$$

This proves claim D.12. □

Now, we prove the lemma by case distinction. First, note that it is easily verified for $D \leq 4$, so we can make use of claim D.12 in the following. Since $(1 + q_1/D) \leq 2$ for any q_1, we only need to show that $\rho' = \prod_{i=2}^{N}(1 + q_i/D) \leq 2$. Moreover, we know, by the premise on subset products, that $\prod_{i=2}^{N} q_i \leq D$.

- Case: $q_2 \leq \sqrt{D}$. Then claim D.12 applies to $q_2, \ldots, q_N$ and yields $\rho' \leq 2$.
- Case: $q_2 > \sqrt{D}$. Then $q_1 q_2 > D$, so $N = 2$, and $\rho' \leq (1 + q_2/D) \leq 2$.

This completes the proof. □

D.6. Proof of theorem 3.3

Proof. Let $\vec{x} \in \mathbb{Z}_p^N$ and $\mu \in \mathbb{Z}_p$. We have to show that if $\vec{x}$ is not uniformly (K', D)-short, i.e. if there is no $d \in [1, D]$ with $d\vec{x} \in [-K', K']_{\mathbb{Z}_p}^N$, then $\Pr\left[\mu + \sum_{i=1}^{N} x_i\gamma_i \in [0, K]_{\mathbb{Z}_p}\right] \leq 8/(D+1)$. Since this must hold for all μ, in particular $-\mu$, it is equivalent to show

$$\Pr\left[\sum_{i=1}^{N} x_i\gamma_i \in [0, K]_{\mathbb{Z}_p} + \mu\right] \leq 8/(D+1).$$

[17] The claim does not hold for $D = 4$, as $(1 + 2/4)^2 = 9/4$. Since $D = 5$ is prime, only $q_1 = 5$, $q_2 = 1$ is possible, but this violates $q_2 \geq 2$, so there is nothing to check. For $D = 6$, choosing $q_1 = 3$, $q_2 = 2$ yields $(1 + \frac{q_1}{D})(1 + \frac{q_2}{D}) = 2$. Moreover, $\frac{q_1+1}{q_1}\frac{q_2+1}{q_2}$ only decreases for larger q_1 or q_2, so the claim holds for $q_1 q_2 > 6$ (with $q_1, q_2 \geq 2$) as well.

[18] Cases with 2 terms were already covered. Cases with more terms, i.e., $N = 3$, still work and are most easily checked programmatically. Cases with 4 or more terms are irrelevant since $2^4 = 16$ already exceeds 15.

We derive this inequality from the core lemma (lemma 3.11). But in order to apply lemma 3.11, we need that all x_i are of the form $x_i \equiv_p \frac{m_i}{d_i}$ with $|m_i| \leq M$ for suitable M. We choose $M = K$, so we have to show $x_i \in \mathbb{Q}_{K,D}$. Thus, we make a case distinction, based on following observation: Consider any fixed choice of $x_1, \ldots x_N \in \mathbb{Z}_p$. Suppose there are two distinct challenges $(\gamma_1, \ldots, \gamma_N)$, $(\gamma'_1, \ldots, \gamma'_N)$ which are accepting and differ only in the i-th component, i.e. $\gamma_j = \gamma'_j$ except for $j = i^*$, and w.l.o.g. $\gamma_{i^*} > \gamma'_{i^*}$. Let $\zeta \equiv_p \sum_{i=1}^N x_i\gamma_i$ and $\zeta' \equiv_p \sum_{i=1}^N x_i\gamma'_i$. Then $\zeta - \zeta' \equiv_p \sum_{i=1} x_i(\gamma_i - \gamma'_i) \equiv_p x_i(\gamma_{i^*} - \gamma'_{i^*})$. Thus $x_{i^*} \equiv_p \frac{\zeta-\zeta'}{\gamma_{i^*}-\gamma'_{i^*}} \in \mathbb{Q}_{K,D}$, since $\zeta - \zeta' \in [-K, K]$ and $\gamma_{i^*} - \gamma'_{i^*} \in [0, D]$. Now, we distinguish two cases.

Case 1: For every i^* there exist two accepting $(\gamma_j)_j \neq (\gamma'_j)_j$ which differ only in the i^*-th component. In that case, we argued above that $x_i \in \mathbb{Q}_{K,D}$ for every i. Thus, lemma 3.11 is applicable with $M \mathrel{\widehat{=}} K$ and D. Moreover, we use that $D(K' + DM + 2) = D(K' + DK + 2) < p$, which is a premise of theorem 3.3 (and also implies $D \cdot (K + 2\beta M) = D \cdot (1 + 2\beta)K < p$). The claim then follows from lemma 3.11. (By choice of the index set I in lemma 3.11, either the common denominator d of the x_is satisfies $D < d < D^2$, in which case eq. (3.12) and $D^2(K' + 1) < p$ implies an error of at most $4/(D + 1)$, or $d \leq D$, in which case eq. (3.13) implies an error of at most $8/(D + 1)$.)

Case 2: The opposite of Case 1, i.e. there exists some i^* for which no two accepting $(\gamma_j)_j \neq (\gamma'_j)_j$ which differ only in the i^*-th component exist. Then $\Pr[\sum_{i\neq i^*} c_i x_i + \gamma_{i^*} x_{i^*} \in [0, K] + \mu] \leq 1/(D + 1)$ for any choice $c_i \in [0, D]$ for $i \neq i^*$. Thus, we get

$$\begin{aligned}
&\Pr\Big[\sum_{i\neq i^*} \gamma_i x_i + \gamma_{i^*} x_{i^*} \in [0, K] + \mu\Big] \\
&= \sum_{c_i\in[0,D], i\neq i^*} \Pr[\forall i \neq i^* \colon \gamma_i = c_i] \cdot \Pr\Big[\sum_{i\neq i^*} c_i x_i + \gamma_{i^*} x_{i^*} \in [0, K] + \mu\Big] \\
&\leq \sum_{c_i\in[0,D], i\neq i^*} \Big(\frac{1}{D+1}\Big)^{N-1} \cdot \frac{1}{D+1} \\
&= \frac{1}{D+1}
\end{aligned}$$

Thus, the probability ε that the test (falsely) accepts satisfies $\varepsilon \leq 1/(D + 1)$. The claim follows. □

Remark D.13 (Compressing the challenge). The verifier's challenge in RAST is relatively large, but it can be compressed. A direct reduction shows that replacing the challenge $\vec{\gamma} \xleftarrow{\$} [0, D]^N$ by $\vec{\gamma} = \mathsf{PRG}(s)$ for $s \xleftarrow{\$} \{0, 1\}^\lambda$, where PRG is a pseudo-random generator, ensures that the soundness error increases only by a negligible amount (assuming PRG is secure). And now, the verifier could send s instead, as a compressed version of $\vec{\gamma} = \mathsf{PRG}(s)$. As the security of RAST is a combinatorial property, it is an interesting question to find small (structured) challenge spaces which are unconditionally secure.

E. Security Reductions

E.1. Security Proof of $\mathsf{Sharp}_{\mathsf{GS}}$

In the following theorem, we show that $\mathsf{Sharp}_{\mathsf{GS}}$ is secure.

Theorem E.1. *The scheme* $\mathsf{Sharp}_{\mathsf{GS}}$ *has correctness error at most* $1 - [(1 - \mathsf{p_r})^3 \cdot (1 - \mathsf{p_x})^{4N}]^R$. *It is non-abort SHVZK under the* SEI *assumption. Suppose now that* $2(B\Gamma^2 + 1)L < p$ *and* $18K^2 < q$ *with* $K = (B\Gamma + 1)L$. *Then it has relaxed soundness under the* DLOG *and* SEI

assumptions in $\mathbb{G}_{\mathsf{com}}$ *and* $\mathbb{G}_{\mathsf{3sq}}$ *with knowledge error* $(\frac{2}{\Gamma+1})^R$ *for the relation*

$$\begin{aligned}\mathsf{R}_{\mathsf{Ext}} =\{(x_1,\ldots,x_N,r)\colon C_x = r_x G_0 + \sum_{i=1}^{N} x_i G_i \\ \wedge\, \exists m_i, d \in \mathbb{Z}\colon x_i \equiv_q \frac{m_i}{d} \wedge -\frac{1}{4B} \leq \frac{m_i}{d} \leq B + \frac{1}{4B} \\ \wedge\, |m_i| \leq (B\Gamma+1)L_x \wedge 1 \leq d \leq \Gamma\}.\end{aligned}$$

Concretely, with the hash-optimization, we have following reductions:

- *For every adversary* $\mathscr{A}$ *against non-abort SHVZK, there are adversaries* $\mathscr{B}_{\mathbb{G}_{\mathsf{com}}}$, $\mathscr{B}_{\mathbb{G}_{\mathsf{3sq}}}$ *whose run-time is roughly that of* $\mathscr{A}$ *and so that* $\mathsf{Adv}^{\text{na-hvzk}}_{\mathscr{A}} \leq \mathsf{Adv}^{\text{sei}}_{\mathbb{G}_{\mathsf{com}},\mathscr{B}_{\mathbb{G}_{\mathsf{com}}}} + R \cdot \mathsf{Adv}^{\text{sei}}_{\mathbb{G}_{\mathsf{3sq}},\mathscr{B}_{\mathbb{G}_{\mathsf{3sq}}}}$.
- *For every adversary* $\mathscr{A}$ *against knowledge which runs at most* T *steps, there are adversaries* $\mathscr{B}_{CR}$, $\mathscr{B}_{\mathbb{G}_{\mathsf{com}}}$, $\mathscr{B}_{\mathbb{G}_{\mathsf{3sq}}}$ *whose* expected *run-time is bounded roughly by* $3 \cdot R \cdot T$, *and so that* $\mathsf{Adv}^{\text{ke}}_{\mathscr{A}} \leq (\frac{2}{\Gamma+1})^R + \mathsf{Adv}^{\text{crhf}}_{\mathsf{Hash},\mathscr{B}_{CR}} + \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}_{\mathsf{com}},\mathscr{B}_{\mathbb{G}_{\mathsf{com}}}} + \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}_{\mathsf{3sq}},\mathscr{B}_{\mathbb{G}_{\mathsf{3sq}}}}$.
- *For witness relation* $\mathsf{R}_{\mathsf{Ext}} \vee \mathsf{R}^{\mathbb{G}_{\mathsf{com}}}_{\text{Bind}} \vee \mathsf{R}^{\mathbb{G}_{\mathsf{3sq}}}_{\text{Bind}} \vee \mathsf{R}_{\text{Coll}}$, *where* $\mathsf{R}^{G}_{\text{Bind}}$ *is a binding-break relation in group* G *(i.e. a non-trivial* DLOG *relation), and* R_{Coll} *is a non-trivial collision for* Hash, *the knowledge error is* $(\frac{2}{\Gamma+1})^R$.

To be precise, we consider the S*-bounded* SEI *assumption in* $\mathbb{G}_{\mathsf{com}}$ *and* $\mathbb{G}_{\mathsf{3sq}}$.

Proof. Throughout this proof, we have $i \in [1,N]$, $j \in [1,3]$, $k \in [1,R]$.

Correctness. As $x_i, y_{i,j} \in [0,B]$ and $\gamma_k \in [1,\Gamma]$, we have $z_{k,i}, z_{k,i,j} \in [0,(B\Gamma+1)L_x]$ (see section 2.5), unless masking aborts. The second check of the verifier succeeds due to the homomorphic properties of MPed and the fact that $f_{k,i} = \gamma_k \cdot x^*_{k,i} + m^*_{k,i}$ by construction.

Honest-verifier zero-knowledge. We define a simulator for valid transcripts as follows. On input of the public parameters and the statement (C_x, B), the simulator Sim proceeds as follows:

- Sample $\gamma_k \xleftarrow{\$} [0,\Gamma]$
- Set $C_y \coloneqq r_y G_0$ for $r_y \xleftarrow{\$} [0,S]$
- Set $C_{k,*} \coloneqq r^*_k H_0$ for $r^*_k \xleftarrow{\$} [0,S]$
- Set $z_{k,i} = \mathsf{mask}_x(0,\widetilde{x}_{k,i})$ and $z_{k,i,j} = \mathsf{mask}_x(0,\widetilde{y}_{k,i,j})$ for $\widetilde{x}_{k,i}, \widetilde{y}_{k,i,j} \xleftarrow{\$} \mathsf{R}_x$
- Set $t_{k,x} = \mathsf{mask}_r(0,\widetilde{r}_{k,x}), t_{k,y} = \mathsf{mask}_r(0,\widetilde{r}_{k,y})$ and $t^*_k = \mathsf{mask}_r(0,\widetilde{r}^*_k)$ for $\widetilde{r}_{k,x}, \widetilde{r}_{k,y}, \widetilde{r}^*_k \xleftarrow{\$} \mathsf{R}_r$
- If any masking failed, then abort, i.e. output $\perp$.
- Compute $D_{k,x} = -\gamma_k C_x + t_{k,x} G_0 + \sum_{i=1}^{N} z_{k,i} G_i$ and $D_{k,y} = -\gamma_k C_y + t_{k,y} G_0 + \sum_{i=1}^{N} \sum_{j=1}^{3} z_{k,i,j} G_{i,j}$
- Compute $f^*_{k,i} = 4 z_{k,i}(\gamma_k B - z_{k,i}) + \gamma_k^2 - \sum_{j=1}^{3} z^2_{k,i,j}$
- Compute $D_{k,*} = -\gamma_k C_{*,k} + t^*_k H_0 + \sum_{i=1}^{N} f^*_{k,i} H_i$
- Set $\Delta = \mathsf{Hash}(\{D_{k,x}, D_{k,y}, D_{k,*}\}_{k=1}^{R})$
- Output $\Delta, C_y, C_{k,*}, \gamma_k, z_{k,i}, z_{k,i,j}, t_{k,x}, t_{k,y}, t_{k,*}$

It is easy to check that the output of Sim is indistinguishable from non-aborting real transcripts. We do so in game hops.

Game 1: Output a transcript tr from an interaction of an honest verifier and prover from the definition of $\mathsf{Sharp}_{\mathsf{GS}}$. If the transcript is aborting, output $\perp$ instead.

Game 2: Act as in *game 1* but instead of computing $D_{k,x}, D_{k,y}$ and $D_{k,*}$ as in the real protocol, compute them as Sim. A quick computation shows that *game 1* and *game 2* are perfectly indistinguishable.

Game 3: Act as in *game 2* but instead of sampling $z_{k,i}, z_{k,i,j}, t_{k,x}, t_{k,y}, t_{k,*}$ as in the real protocol, sample them as Sim, i.e. via $\mathsf{mask}(0,\cdot)$. The games *game 2* and *game 3* are statistically indistinguishable. Namely, their statistical distance is bounded by $RN(1+3)\varepsilon_x + R\varepsilon_r$, where the masking errors ε_x and ε_r correspond to the masking schemes mask_x and mask_r (see section 2.5). Due to uniform rejection sampling, $\varepsilon_x = \varepsilon_r = 0$. (Note that, if, say $z_{k,i} = \perp$ we cannot define the corresponding $D_{k,x}$. This is not a problem, since we consider non-abort SHVZK, hence a transcript where $z_{k,i} = \perp$ is replaced by $\perp$, both in game 2 and 3.)

Game 4: Instead of computing the commitments $C_y, C_{k,*}$ as in the real protocol, compute them as Sim. *Game 3* and *game 4* are indistinguishable under the hiding property of the commitment scheme. More precisely, the we need 1 reduction to the SEI assumption in $\mathbb{G}_{\mathsf{com}}$ for C_y, and R in $\mathbb{G}_{\mathsf{3sq}}$ for $C_{k,*}$.

As the output of *game 4* is equal to the output of Sim, $\mathsf{Sharp}_{\mathsf{GS}}$ is non-abort SHVZK.

Soundness. We assume that we are given a number of accepting related transcripts, and first show the statement for a single repetition, i.e. $R = 1$. After that, we discuss how repetitions change the security proofs and how to obtain the transcripts. For readability, we omit k (denoting the current repetition) in the following as index from the transcripts. Assume that a PPT adversary can interactively produce three valid transcripts tr, tr', tr'' with fixed first message Δ, C_y, C_*, and distinct challenges γ, γ', γ'' and masked terms $[z_i, z_{i,j}, t_x, t_y, t_*]$, $[z'_i, z'_{i,j}, t'_x, t'_y, t'_*]$ and $[z''_i, z''_{i,j}, t''_x, t''_y, t''_*]$. We define F_x, F'_x, F''_x as in the verification, similarly for F_y, F_*, f_i^*. We denote by $\overline{X}$ and $\underline{X}$ the differences $X' - X$ and $X'' - X$ respectively for $X \in [\gamma_k, z_i, z_{i,j}, t_x, t_y, t_*, F_x, F_y, F_*, f_i]$. Without loss of generality, $\overline{\gamma}, \underline{\gamma} > 0$. Note that $p = \mathsf{ord}(\mathbb{G}_{\mathsf{com}})$ and $q = \mathsf{ord}(\mathbb{G}_{\mathsf{3sq}})$.

Step 1 – Opening the Commitments: First, we extract openings of C_x, C_y, C_*. By collision resistance of Hash, we have $D_x := F_x = F'_x = F''_x$. Further, the check of the verifier guarantees:

$$\begin{aligned} D_x &= -\gamma C_x + t_x G_0 + \sum_{i\in[1,N]} z_i G_i \\ &= -\gamma' C_x + t'_x G_0 + \sum_{i\in[1,N]} z'_i G_i \end{aligned}$$

Rearranging this equation leads to the following equality:

$$\begin{aligned} \overline{\gamma} C_x &= \overline{t_x} G_0 + \sum_{i\in[1,N]} \overline{z_i} G_i \\ \implies C_x &= \overline{t_x}/\overline{\gamma} G_0 + \sum_{i\in[1,N]} \overline{z_i}/\overline{\gamma} G_i. \end{aligned}$$

Thus, C_x commits to values $x_i \equiv_p \overline{z_i}/\overline{\gamma} \in \mathbb{Z}_p$. With the same calculation, we can show that $x_i \equiv_p \underline{z_i}/\underline{\gamma}$. Note that x_i is well defined as MPed is binding. (We reduce to DLOG in $\mathbb{G}_{\mathsf{com}}$ and

$\mathbb{G}_{\mathsf{3sq}}$ for binding.[19]) Similarly, we find openings for the remaining commitments

$$C_y = \overline{t_y}/\overline{\gamma}G_0 + \sum_{i\in[1,N],j\in[1,3]} \overline{z_{i,j}}/\overline{\gamma}G_{i,j} \text{ and}$$
$$C_* = \overline{t^*}/\overline{\gamma}H_0 + \sum_{i\in[1,N]} \overline{f_i^*}/\overline{\gamma}H_i.$$

So C_y commits to values $y_{i,j} \equiv_p \overline{z_{i,j}}/\overline{\gamma} \equiv_p \underline{z_{i,j}}/\underline{\gamma} \in \mathbb{Z}_p$ and C_* to $x_* \equiv_q \overline{f_i^*}/\overline{\gamma} \equiv_q \underline{f_i^*}/\underline{\gamma} \in \mathbb{Z}_q$.

Step 2 – Decomposition: We now show that the three-square decomposition holds and that $[x_i]_{\mathbb{Q}}$ is indeed in the range $[-\frac{1}{4B}, B+\frac{1}{4B}]_{\mathbb{Q}}$. We proceed similarly to [Cou+21a] but our proof is more subtle, as we argue over two different groups with incompatible (prime) modulus. Nonetheless, we can conclude $[x_i]_{\mathbb{Q}} \in [-\frac{1}{4B}, B+\frac{1}{4B}]_{\mathbb{Q}}$ since the rational representative is unique in both groups.

First, we define $\widehat{x_i} \equiv_q \overline{z_i}/\overline{\gamma} \equiv_q \underline{z_i}/\underline{\gamma}$. Note that $\widehat{x_i}$ is well-defined as all values are short and thus

$$\begin{aligned}\overline{z_i}/\overline{\gamma} \equiv_p \underline{z_i}/\underline{\gamma} &\implies \overline{z_i}\underline{\gamma} \equiv_p \underline{z_i}\overline{\gamma} \implies \overline{z_i}\underline{\gamma} = \underline{z_i}\overline{\gamma} \text{ over } \mathbb{Z}\\ &\implies \overline{z_i}\underline{\gamma} \equiv_q \underline{z_i}\overline{\gamma} \implies \overline{z_i}/\overline{\gamma} \equiv_q \underline{z_i}/\underline{\gamma}.\end{aligned}$$

Now, we set $m_i \equiv_q z_i - \gamma\widehat{x_i}$. Using the definition of $\widehat{x_i}$, we have

$$\begin{aligned}m_i &\equiv_q z_i - \gamma\widehat{x_i} \equiv_q z_i + z_i' - z_i' - \gamma\widehat{x_i} \equiv_q -\overline{z_i} + z_i' - \gamma\widehat{x_i}\\ &\equiv_q \overline{\gamma}\widehat{x_i} + z_i' - \gamma\widehat{x_i} \equiv_q (\gamma'-\gamma)\widehat{x_i} + z_i' - \gamma\widehat{x_i} \equiv_q z_i' - \gamma'\widehat{x_i}.\end{aligned}$$

Also, $m_i \equiv_q z_i'' - \gamma''\widehat{x_i}$ follows accordingly. We similarly set $\widehat{x_{i,j}} \equiv_q \overline{z_{i,j}}/\overline{\gamma}$ and $m_{i,j} \equiv_q z_{i,j} - \gamma\widehat{x_{i,j}} \equiv_q z_{i,j}' - \gamma'\widehat{x_{i,j}} = z_{i,j}'' - \gamma''\widehat{x_{i,j}}$, where the equalities follow as above. Inserting these equalities and interpreting f_i^* (similarly $(f_i^*)', (f_i^*)''$) as a polynomial with variable γ, we obtain:

$$\begin{aligned}f_i^* &\equiv_q \gamma^2[4\widehat{x_i}(B-\widehat{x_i}) + 1 - \sum_{j\in[1,3]} \widehat{x_{i,j}}^2] + \gamma\alpha_1 + \alpha_0,\\ (f_i^*)' &\equiv_q (\gamma')^2[4\widehat{x_i}(B-\widehat{x_i}) + 1 - \sum_{j\in[1,3]} \widehat{x_{i,j}}^2] + \gamma'\alpha_1 + \alpha_0\\ (f_i^*)'' &\equiv_q (\gamma'')^2[4\widehat{x_i}(B-\widehat{x_i}) + 1 - \sum_{j\in[1,3]} \widehat{x_{i,j}}^2] + \gamma''\alpha_1 + \alpha_0\end{aligned}$$

for some appropriate α_1, α_0. We can subtract the first from the second (third) equation and then divide the resulting equation by $\overline{\gamma}$ ($\underline{\gamma}$) respectively. This leads to:

$$\begin{aligned}\overline{f_i^*}/\overline{\gamma} &\equiv_q (\gamma'+\gamma)[4\widehat{x_i}(B-\widehat{x_i}) + 1 - \sum_{j\in[1,3]} \widehat{x_{i,j}}^2] + \alpha_1,\\ \underline{f_i^*}/\underline{\gamma} &\equiv_q (\gamma''+\gamma)[4\widehat{x_i}(B-\widehat{x_i}) + 1 - \sum_{j\in[1,3]} \widehat{x_{i,j}}^2] + \alpha_1.\end{aligned}$$

As $x_* \equiv_q \overline{f_i^*}/\overline{\gamma} \equiv_q \underline{f_i^*}/\underline{\gamma}$ (see first step), we obtain:

$$\begin{aligned}&(\gamma''-\gamma')[4\widehat{x_i}(B-\widehat{x_i}) + 1 - \sum_{j\in[1,3]} \widehat{x_{i,j}}^2] \equiv_q 0\\ \implies &4\widehat{x_i}(B-\widehat{x_i}) + 1 \equiv_q \sum_{j\in[1,3]} \widehat{x_{i,j}}^2\\ \implies &4\overline{z_i}(\overline{\gamma}B - \overline{z_i}) + \overline{\gamma}^2 \equiv_q \sum_{j\in[1,3]} \overline{z_{i,j}}^2\end{aligned}$$

[19] Note that due to random self-reducibility of DLOG, we need not incur a loss of N in the reduction.

Setting $K = (B\Gamma+1)L_x$ and noting $|\overline{z_i}| \leq K$, we find $|4\overline{z_i}(\overline{\gamma}B - \overline{z_i}) + \overline{\gamma}^2| \leq 4K^2 + 4K^2 + K^2 < q/2$ and $\sum_{j\in[1,3]} \overline{z_{i,j}}^2 \leq 3K^2 < q/2$. Thus, the equation holds over the integers and as a result, it holds that $4\overline{z_i}(\overline{\gamma}B - \overline{z_i}) + \overline{\gamma}^2 \geq 0$. Dividing by $\overline{\gamma}$ yields $4\frac{\overline{z_i}}{\overline{\gamma}} i(B - \frac{\overline{z_i}}{\overline{\gamma}}) + 1 \geq 0$. Now, lemma B.2 implies that $\frac{\overline{z_i}}{\overline{\gamma}} \in [-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}}$.

Step 3 – Repetitions: Consider a setting with repetitions. Suppose we are given three related transcripts such that in repetition k, the challenges $\gamma_k, \gamma'_k, \gamma''_k$ are pairwise distinct. Then the previous steps apply, and we conclude the same soundness guarantees. Note that it suffices to have such related transcripts for *any* of the k repetitions. Further, since the *same* commitment C_x is used in all iterations, extractions $x_{k,i}$ (of x_i) for differing k must coincide, or the binding property and hence DLOG is broken in $\mathbb{G}_{\mathsf{com}}$.

Step 4 – Obtaing the transcripts: It is well-known how to obtain related transcripts, but we give a brief sketch for the sake of completeness. First, run the (malicious) prover with a random challenge. If the honest verifer rejects, the extractor has nothing to do; it just outputs this view. So assume otherwise. If $R = 1$, rewind the (malicious) prover and try (fresh) random challenges (without repetition) until 3 transcripts are found or all challenges exhausted. For $R > 1$, a very naive strategy exploits that the protocol is $(2^R + 1)$-special sound, but this degrades the knowledge error. A less wasteful approach works with about $3R$ expected rewinds. The basic idea is to not pick all challenges γ_k fresh, but keep all but one fixated, and do that for all $k = 1, \ldots, R$ in parallel. See [Bau+18a; ACK21] for concrete examples. □

E.2. Proof of $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$

In this section, we prove the security of $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$. As usual, we consider the optimized variant which uses a CRHF.

Theorem E.2. *The scheme* $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ *has correctness error at most* $1 - (1 - 1/L)^{3+2R+4N}$. *It is non-abort SHVZK under the* SEI *assumption. Let* $K' = (1 + 2\beta)K$ *where* $K = (B\Gamma + 1)L$ *and* $\beta = \min(4N, \mathsf{primlmin}(\Gamma + 1))$. *Suppose* $9(K')^2 < q/2$ *and* $(\Gamma + 1)^R - 1 < p$. *Then* $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ *has relaxed soundness under the* DLOG *and* SEI *in* $\mathbb{G}_{\mathsf{com}}$ *and* $\mathbb{G}_{\mathsf{3sq}}$ *with knowledge error* $\frac{2+8^R}{(\Gamma+1)^R}$ *for relation*

$$\begin{aligned}\mathsf{R}_{\mathsf{Ext}} = \{&(x_1, \ldots, x_N, r) \colon C_x = r_x G_0 + \sum_{i=1}^{N} x_i G_i \\ &\wedge \exists m_i, d \in \mathbb{Z} \colon x_i \equiv_q \frac{m_i}{d} \wedge -\frac{1}{4B} \leq \frac{m_i}{d} \leq B + \frac{1}{4B} \\ &\wedge |m_i| \leq K' \wedge 1 \leq d \leq \Gamma\}.\end{aligned}$$

Concretely, with the hash-optimization, we have following reductions:

- *For every adversary* $\mathscr{A}$ *against non-abort SHVZK, there are adversaries* $\mathscr{B}_{\mathbb{G}_{\mathsf{com}}}$, $\mathscr{B}_{\mathbb{G}_{\mathsf{3sq}}}$ *whose run-time is roughly that of* $\mathscr{A}$ *and so that* $\mathsf{Adv}^{\text{na-hvzk}}_{\mathscr{A}} \leq \mathsf{Adv}^{\text{sei}}_{\mathbb{G}_{\mathsf{com}}, \mathscr{B}_{\mathbb{G}_{\mathsf{com}}}} + R \cdot \mathsf{Adv}^{\text{sei}}_{\mathbb{G}_{\mathsf{3sq}}, \mathscr{B}_{\mathbb{G}_{\mathsf{3sq}}}}$.
- *For every adversary* $\mathscr{A}$ *against knowledge which runs at most* T *steps, there are adversaries* $\mathscr{B}_{CR}$, $\mathscr{B}_{\mathbb{G}_{\mathsf{com}}}$, $\mathscr{B}_{\mathbb{G}_{\mathsf{3sq}}}$ *whose* expected *run-time is bounded roughly by* $6 \cdot R \cdot T$, *and so that* $\mathsf{Adv}^{\text{ke}}_{\mathscr{A}} \leq \frac{2+8^R}{(\Gamma+1)^R} + \mathsf{Adv}^{\text{crhf}}_{\mathsf{Hash}, \mathscr{B}_{CR}} + \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}_{\mathsf{com}}, \mathscr{B}_{\mathbb{G}_{\mathsf{com}}}} + \mathsf{Adv}^{\text{dlog}}_{\mathbb{G}_{\mathsf{3sq}}, \mathscr{B}_{\mathbb{G}_{\mathsf{3sq}}}}$.
- *For witness relation* $\mathsf{R}_{\mathsf{Ext}} \vee \mathsf{R}^{\mathbb{G}_{\mathsf{com}}}_{\text{Bind}} \vee \mathsf{R}^{\mathbb{G}_{\mathsf{3sq}}}_{\text{Bind}} \vee \mathsf{R}_{\text{Coll}}$, *where* $\mathsf{R}^{G}_{\text{DL-rel}}$ *is a non-trivial DLOG relation in group* G, *and* R_{Coll} *is a non-trivial collision for* Hash, *the knowledge error is* $\frac{2+8^R}{(\Gamma+1)^R}$.

The rest of this section consists of a proof of theorem E.2.

Correctness. Correctness follows by a straightforward check. Whenever an honest prover does not abort (due to masking), the honest verifier will accept.

Non-abort SHVZK This follows almost as for $\mathsf{Sharp}_{\mathsf{GS}}$ in theorem E.1. Namely, Phase 2 can be argued identically (except, that there are no repetitions now). Once Phase 2 is replaced by a simulation, Phase 1 can be simulated by using $x_i = 0$ instead of the real witness. Since ζ_k are masked terms, this incurs k masking errors, which however are 0 for uniform rejection sampling.

Soundness. The rest of this section is dedicated to proving the soundness error.

E.2.1. Step 1: Extracting Phase 2.

We begin as described in the outline. Let G_0 be the knowledge soundness game from definition A.21. As the first step, define G_1 which only differs from G_0 by replacing the malicious prover with an extractor in Phase 2. More concretely, note that Phase 2 is a 3-special sound Σ-protocol for the relation (where $y_{i,0} \coloneqq x_i$)

$$\begin{aligned}\mathsf{R}_{\mathsf{Ext}} = \big\{&(C_x, C_y, \{\zeta\}_{k\in[1,R]}; \{x_i\}_{i\in[1,N]}, \{y_{i,j}\}_{i\in[1,N],j\in[0,3]},\\ &r_x, r_y, \{\mu_k\}_{k\in[1,R]}) \colon\\ &C_x = r_x G_0 + \sum_{i=1}^{N} x_i G_i\\ &\wedge\, \forall\, k \in [1,R] \colon \sum_{i=1}^{N}\sum_{j=0}^{3} y_{i,j}\gamma_{i,j} + \mu_k = \zeta_k\\ &\wedge\, C_y = r_y G_0 + \sum_{i=1}^{N}\sum_{j=1}^{3} y_{i,j} G_{i,j} + \sum_{k=1}^{R} \mu_k \widetilde{G}_k\\ &\wedge\, \forall i \in [1,N] \colon\ 1 + 4x_i(B - x_i) \equiv_q \sum_{j\in[1,3]} y_{i,j}^2\big\}.\end{aligned}$$

or a hash-collision or DLOG relation, i.e. $\mathsf{R}_{\mathsf{Ext}} \vee \mathsf{R}_{\mathrm{Bind}}^{\mathbb{G}_{\mathsf{com}}} \vee \mathsf{R}_{\mathrm{Bind}}^{\mathbb{G}_{\mathsf{3sq}}} \vee \mathsf{R}_{\mathrm{Coll}}$. This follows analogously to theorem E.1 for $\mathsf{Sharp}_{\mathsf{GS}}$, up to standard changes. Thus, as in theorem E.1, we find that, the run-time changes from strict run-time t_0 to *expected* time $t_1 \approx 3t_0$ and the knowledge error is $2/(\Gamma+1)^R$. In game G_1, we return 1 iff the extraction succeeded as well. Overall, it follows that

$$\Pr[\mathsf{G}_0 = 1] \leq \Pr[\mathsf{G}_1 = 1] + 2/(\Gamma+1)^R \quad \text{and} \quad t_1 \approx 3t_0.$$

E.2.2. Step 2: Extracting Phase 1.

Recall that Phase 2 of the protocol is actually a proof of knowledge for $\mathsf{R}_{\mathsf{Ext}}$ with statement $(C_x, C_y, \{\zeta_k\}_k)$. In game G_1, we always try to extract Phase 2, so now, we are almost in the setting of (random affine) shortness testing. The main difference is, that in the latter setting, the choice of $(\{x_i\}, \{y_{i,j}\})$ would be *fixed beforehand*, whereas in our case, it is only *committed to*. Thus, we need to account for the case of a binding break.

Looking ahead, the completed extractor works as follows:

1. Pick a uniform challenge $\gamma_{i,j}^{(k)} \xleftarrow{\$} [0,\Gamma]$ ($i \in [1,N]$, $j \in [0,3]$, $k \in [1,R]$) for Phase 1 and run the extractor for Phase 2.
2. If extraction (of Phase 2) fails, also output failure, i.e. $\perp_{\mathsf{ext}}$.
3. If the verifier did not accept the run, output the generated view. (There is nothing to do.)

4. If the extracted witness $(y_{i,j})_{i,j}$ (where $y_{i,0} = x_i$) is of the form $y_{i,j} = \frac{m_{i,j}}{d}$ for $d \in [1, \Gamma]$ and $m_i \in [0, K']$ with $K' = (1 + 2\beta K)$, output $(x_i)_i$.[20] In this case, $x_i = \frac{m_i}{d} \in [0, B]$ as claimed.

5. Else, the extracted $(x_i)_i$ are "invalid". Try to obtain a DLOG relation as follows:
   - Rewind before sending the challenge (in Phase 1) and pick fresh uniform challenges $\widetilde{\gamma}_{i,j}^{(k)} \xleftarrow{\$} [0, \Gamma]$ (with repetition) for Phase 1 and run the extractor for Phase 2. Repeat until again a witness $(y'_{i,j})_{i,j}$ is output.
   - If $(y_{i,j})_{i,j} \neq (y'_{i,j})_{i,j}$, return the non-trivial DLOG relation corresponding to $(y_{i,j} - y'_{i,j})_{i,j}$.
   - Otherwise, output failure, i.e. $\perp_{\mathsf{ext}}$.

Let us now analyze this extraction and the soundness of the protocol. Extraction failure (item 2) and verifier rejection (item 3) are trivial to account for. So let us assume that a witness $(\{x_i\}, \{y_{i,j}\})$ was extracted from Phase 2. Each test $\sum_{i=1}^{N} \sum_{j=0}^{3} \gamma_i^{(k)} y_{i,j} + \mu_k \in [0, K]$ (where $y_{i,0} = x_i$) is a random affine shortness test (definition 3.2), where μ_k plays the role of the constant offset μ and $4N$ elements are checked at once. The parameters of this test are dimension $N \mathrel{\widehat{=}} 4N$, and range bound K, test distribution $U_{[0,\Gamma]^{4N}}$ and offset μ_k. As shown in theorem 3.3, the test is fractional (K', Γ)-sound with error $\kappa \leq 8/(\Gamma + 1)$, where $K' = (1 + 2\beta)K$ and $\beta = \min(4N, \mathsf{primlmin}(\Gamma + 1))$. The probability to cheat (in all of them) is therefore $\kappa^R \leq (\frac{8}{\Gamma+1})^R$. Now there are two cases:

Case 1 (item 4): There exists some $d \in [1, \Gamma]$ such that $dy_{i,j} \in [-K', K']$ for all $i \in [1, N]$, $j \in [0, 3]$, where $y_{i,0} \coloneqq x_i$. For this case, consider the quadratic relations in Phase 2, which are known to hold over $\mathbb{Z}_q$ (for all i):

$$\begin{aligned} 1 + 4x_i(B - x_i) &\equiv_q \sum_{j \in [1,3]} y_{i,j}^2 \\ \iff d^2 + 4dx_i(dB - dx_i) &\equiv_q \sum_{j \in [1,3]} (dy_{i,j})^2 \end{aligned}$$

Since and $dy_{i,j} \in [-K', K']$ (where $y_{i,0} = x_i$), the left-hand side has absolute value at most $\Gamma^2 + 4K'(B\Gamma + K') \leq 9(K')^2 < q/2$. The right-hand-side is at most $3(K')^2 < q/2$. Thus, the right equality holds over the integers, and the left equality holds over the rationals. Consequently, we find $x_i \in [-\frac{1}{4}B, B + \frac{1}{4}B]_{\mathbb{Q}}$ (by lemma B.2). Since additionally $dx_i \in [-K', K']$ for all i, we have found a witness for $\mathsf{R}_{\mathsf{Ext}}$, which completes this case.

Case 2 (item 5): There is no $d \in [1, \Gamma]$ such that $dy_{i,j} \in [-K', K']$ for all i, j, but the shortness tests failed to catch this. In this case, the extractor rewinds and retries Phase 1 (with fresh challenges and still running the extractor for Phase 2) until a second extracted run is found; denote the extracted witness by $(y'_{i,j})_{i,j}$. Let $\varepsilon_1 = \Pr[\mathsf{G}_1 = 1]$. It is easy to check that the expected number of retries is 1 and that, overall, the expected time t_2 for the extractor is roughly bounded by $2t_1 \leq 6t_0$. Since the soundness error of the repeated shortness test is κ^R, with probability at least $(\varepsilon_1 - \kappa^R)/\varepsilon_1$, it happens that $(y_{i,j})_{i,j} \neq (y'_{i,j})_{i,j}$ or $\mu_k \neq \mu'_k$ for the second accepting transcipt. In that case, a non-trivial DLOG relation can be derived from the binding break, i.e. the two different witnesses.[21]

Define G_2 as a run of the complete extractor, and let it output 1 if and only if the verifier was convinced in the initial run and a valid witness is outputted. Note that G_0 resp. G_2 now correspond

[20] Note that we need an efficient algorithm to check this. But as noted in remark 2.7, for any choice of $M, D \in \mathbb{N}$ with $MD < p/2$ we can efficiently [FSW03] compute (m, d) for $x \equiv_p \frac{m}{d}$ if $x \in \mathbb{Q}_{M,D}$. In our setting, $M = K'$ and $D = \Gamma$ satisfies $\Gamma K' < q/2$ by assumption.

[21] The complete extracted witnesses $\vec{w}, \vec{w}'$ also contain components r_x, r_y.

to the real resp. ideal executions in the definition of knowledge soundness (definition A.21). We see that, except with probability at most $\varepsilon_1 \cdot \frac{\kappa^R}{\varepsilon_1} = \kappa^R$, where $\varepsilon_1 = \Pr[\mathsf{G}_1 = 1]$, game G_2 succeeds in producing a valid witness $\mathsf{R}_{\mathsf{Ext}} \vee \mathsf{R}^{\mathbb{G}_{\mathsf{com}}}_{\mathrm{Bind}} \vee \mathsf{R}^{\mathbb{G}_{\mathsf{3sq}}}_{\mathrm{Bind}} \vee \mathsf{R}_{\mathrm{Coll}}$. Thus, overall we find

$$\Pr[\mathsf{G}_2 = 1] \leq \Pr[\mathsf{G}_1 = 1] + \kappa^R \leq \Pr[\mathsf{G}_0 = 1] + 2/(\Gamma+1)^R + \kappa^R$$

If the extractor does not fail, it returns a witness for $\mathsf{R}_{\mathsf{Ext}} \vee \mathsf{R}^{\mathbb{G}_{\mathsf{com}}}_{\mathrm{Bind}} \vee \mathsf{R}^{\mathbb{G}_{\mathsf{3sq}}}_{\mathrm{Bind}} \vee \mathsf{R}_{\mathrm{Coll}}$, where $\mathsf{R}^G_{\mathrm{Bind}} = \mathsf{R}^G_{\text{DL-rel}}$ is a binding break, i.e. a non-trivial DLOG relation in $G \in \{\mathbb{G}_{\mathsf{com}}, \mathbb{G}_{\mathsf{3sq}}\}$, and $\mathsf{R}_{\mathrm{Coll}}$ is a non-trivial collision for Hash. The knowledge error for this witness relation is $2/(\Gamma+1)^R + \kappa^R \leq (2+8^R)/(\Gamma+1)^R$, as claimed in last item of theorem E.2. Witnesses for $\mathsf{R}^{\mathbb{G}_{\mathsf{com}}}_{\mathrm{DLog}}$, $\mathsf{R}^{\mathbb{G}_{\mathsf{3sq}}}_{\mathrm{DLog}}$ and $\mathsf{R}_{\mathrm{Coll}}$ can instead be viewed as adversaries against DLOG and collision resistance, showing the second item. This completes the proof of knowledge soundness.

E.3. Security Proof of $\mathsf{Sharp}_{\mathsf{HO}}$

Here, we prove the security of theorem C.1.

Proof. Here, we demonstrate correctness, soundness and zero-knowledge of $\mathsf{Sharp}^{+\mathsf{HO}}_{\mathsf{GS}}$ in more detail. We note that all assumptions are w.r.t. to Sample, in particular, we assume the adversary has access to the random coins ρ_i used to generate the hidden order group elements in the CRS. (This is not the case for elements of $\mathbb{G}_{\mathsf{com}}$ and $\mathbb{G}_{\mathsf{3sq}}$. There, we still assume invertible sampling.)

Correctness. The rejection probability is increased by a factor of $(1 - \mathsf{p_r}') \leq (1 - 1/L)$ due to the additional masking of t'_x. It is straightforward to see that all "old" checks will pass, as the computations and checks for z_i, F_i are unmodified. The modified computation of the hash of Δ will pass, if $D'_x = F'_x$ holds. Hence, it remains to show that $D'_x = F'_x$, i.e.

$$\begin{aligned} D'_x &= \widetilde{r}'_x G'_0 + \sum_{i=1}^{N} (\Gamma+1)^{k-1} \widetilde{x}'_i G'_i \\ &\overset{!}{=} -\gamma' C'_x + t'_x \cdot G'_0 + \sum_{i \in [1,N]} z'_i \cdot G'_i \\ &= F'_x \end{aligned}$$

holds, where $\gamma' = \sum_{k=1}^{R} \gamma_k (\Gamma+1)^{k-1} \in [0, (\Gamma+1)^R - 1]$, and $\widetilde{x}'_i = \sum_{k=1}^{R} (\Gamma+1)^{k-1} \widetilde{x}_{k,i}$ and $z'_i = \sum_{k=1}^{R} (\Gamma+1)^{k-1} \cdot z_{k,i}$. We have

$$\begin{aligned} F'_x &= -\gamma' C'_x + t'_x \cdot G'_0 + \sum_{i \in [1,N]} z'_i \cdot G'_i \\ &= -\gamma' C'_x + (\gamma' r'_x + \widetilde{r}'_x) \cdot G'_0 + \sum_{i \in [1,N]} z'_i \cdot G'_i \\ &= \widetilde{r}'_x \cdot G'_0 + (-\gamma' C'_x + \gamma' r'_x + \sum_{i \in [1,N]} z'_i \cdot G'_i) \end{aligned}$$

Plugging in $\gamma' = \sum_{k=1}^{R} (\Gamma+1)^{k-1} \cdot \gamma_k$, we find that

$$\begin{aligned}
&- \gamma' C'_x + \gamma' r'_x + \sum_{i \in [1,N]} z'_i \cdot G'_i \\
&= \sum_{k=1}^{R} (\Gamma+1)^{k-1} \left(-\gamma_k C'_x + \gamma_k r'_x \cdot G'_0 + \sum_{i \in [1,N]} z_{k,i} \cdot G'_i \right) \\
&= \sum_{k=1}^{R} (\Gamma+1)^{k-1} \sum_{i=1}^{N} (-\gamma_k x_i + z_{k,i}) G'_i \\
&= \sum_{k=1}^{R} (\Gamma+1)^{k-1} \sum_{i=1}^{N} \widetilde{x}_{k,i} G'_i
\end{aligned}$$

since by construction $C'_x = r'_x G_0 + \sum_{i=1}^{N} x_i G'_i$ and $z_{k,i} = \gamma_k x_i + \widetilde{x}_{k,i}$. Thus,

$$F'_x = \widetilde{r}'_x G'_0 + \sum_{k=1}^{R} \sum_{i=1}^{N} (\Gamma+1)^{k-1} \widetilde{x}_{k,i} G'_i = D'_x.$$

Soundness. The argument for soundness of $\mathsf{Sharp}_{\mathsf{GS}}^{+\mathsf{HO}}$ is basically the same as for $\mathsf{Sharp}_{\mathsf{GS}}$ in theorem E.1, except, that the properties of the MPed commitment in $\mathbb{H}$ must be exploited additionally.

Let $\widehat{\Gamma} = (\Gamma+1)^k - 1$. Observe that, by construction, the synthetic challenge γ' is uniform in $[0, \widehat{\Gamma}]$. Moreover, the synthesized proof of short opening is almost the same of the *simple PoSO*[22], and our soundness argument as well, with the only difference being the choice of masking. Namely, the distributions of the mask $\widetilde{x}'_i$ is not the usual one. However, for soundness, the distribution of the mask does not matter at all (it may be adversarially chosen anyway). Consequently, the argument for the PoSO in theorem E.1 applies without change. That is, either two[23] accepting transcripts tr and $\widehat{tr}$ with same first message but different challenges yield witnesses $x'_i = \frac{z'_i - \widehat{z'_i}}{\gamma' - \widehat{\gamma'}}$ of the form $x'_i = a_i / 2^{e_i}$ for $e_i \geq 0$, $a_i \in \mathbb{Z}$ or a (Γ, e)-relaxed DLOG relation in $\mathbb{H}$ was found. By assumption, finding a (Γ, e, N)-relaxed DLOG relation w.r.t. Sample is hard. (Note we use a hat $\widehat{\cdot}$ to distinguish the transcripts, since primes $\cdot'$ are already used to indicate elements of our augmentation.)

Recall that we argued in particular, that in each iteration,

- either $\gamma_k = \widehat{\gamma}_k$ and $z_{k,i} = \widehat{z}_{k,i}$, i.e. this repetition "does not extract", or
- we extract $x_{k,i}$ and $x_i = x_{k,i}$ is unique for all "extracted" repetitions k.

or a non-trivial DLOG relation was found. Now, we have to show that $x'_i = x_i$, i.e. the extracted witness of the synthesized hidden order proof of small opening coincides with the other extractions. For this, note that

$$\begin{aligned}
x'_i = \frac{z'_i - \widehat{z'}_i}{\gamma' - \widehat{\gamma'}} &= \frac{\sum_{k=1}^{R} (\Gamma+1)^{k-1} (z_{k,i} - \widehat{z}_{k,i})}{\sum_{k=1}^{R} (\Gamma+1)^{k-1} (\gamma_k - \widehat{\gamma}_k)} \\
&= \frac{\sum_{k=1}^{R} (\Gamma+1)^{k-1} (x_{k,i} (\gamma_k - \widehat{\gamma}_k))}{\sum_{k=1}^{R} (\Gamma+1)^{k-1} (\gamma_k - \widehat{\gamma}_k)} \\
&= x_i \cdot \frac{\sum_{k=1}^{R} (\Gamma+1)^{k-1} (\gamma_k - \widehat{\gamma}_k)}{\sum_{k=1}^{R} (\Gamma+1)^{k-1} (\gamma_k - \widehat{\gamma}_k)} = x_i
\end{aligned}$$

[22] That is, usual the Σ-protocol for opening with short challenge and shortness check, as used in $\mathsf{Sharp}_{\mathsf{GS}}$ for example.

[23] To show that x_i is of the form $x' = a/2^e$, two transcripts suffice. The soundness of the full argument still needs three transcripts.

where we used that $x_{k,i} = x_i$ for all k.[24] Thus, $x'_i = x_i$ and the extracted witnesses of all repetitions coincide. This finishes the proof.

For $\mathsf{Sharp}^{+\mathsf{HO}}_{\mathsf{PoSO}}$, an analogous reasoning applies, though simpler since "synthesized" variables are not needed.

Non-abort SHVZK The simulator works as the simulator in theorem E.1 (resp. theorem E.2), with following additional steps:

1. Compute $\gamma' = \sum_{k=1}^{R} \gamma_i (\Gamma + 1)^{k-1}$.
2. Set $C'_x \xleftarrow{\$} \mathbb{H}$.
3. Let $z'_i = \sum_{k=1}^{R} (\Gamma + 1)^{k-1} \cdot z_{k,i}$ (using the simulated $z_{k,i}$).
4. Set $t'_x = \mathsf{mask}_{r'}(0, \widetilde{r}')$.
5. If masking fails, then abort, i.e. output $\perp$.
6. Compute $D'_x = -\gamma' C'_x + t'_x G'_0 + \sum_{i=1}^{N} z'_i G'_i$.
7. Adapt the output to include the additional messages.

It is easy to check that the output is indistinguishable from non-aborting real transcripts. The justification is almost identical to the one in theorem E.1. Namely, starting from the honest computation, first compute D'_x is in step 6 above (with otherwise honest values). This change is only conceptual. Then, compute t'_x as in step 4 above. Finally, an additional step is required to justify the switch from computing $C'_x = r'_x G'_0 + \sum_{i=1}^{N} x_i G'_i$ to $C'_x \xleftarrow{\$} \mathbb{H}$. Since r'_x is not used anymore, we can reduce this to SI and SEI assumptions (w.r.t. Sample). By SEI we can replace the term $A = r'_x G'_0$ by $A \xleftarrow{\$} \langle G'_0 \rangle$. Then, by SI we can replace $A \xleftarrow{\$} \langle G'_0 \rangle$ by $A \xleftarrow{\$} \mathbb{H}$. Now, C'_x is uniform distributed in $\mathbb{H}$. So we can sample $C'_x \xleftarrow{\$} \mathbb{H}$ instead. This is done by the simulator in step 2, and indeed, this game is the simulation, completing the proof. □

F. Additional Tables

Here, we provide some tables with an overview of the parameters and proof sizes of $\mathsf{Sharp}_{\mathsf{GS}}$ in table 3 and $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ in table 4.

[24] Strictly speaking, if $\gamma_k = \widehat{\gamma}_k$, then $x_{k,i}$ is not defined. By our assumption, we may assume it exists and equals x_i. This is a mere simplification, as the contribution of repetition k to the sum is 0 anyway, since (again by assumption) $z_{k,i} - \widehat{z}_{k,i} = 0$.

Table 3: Overview of parameters (where $S = 2^{256} - 1$ always) and proof sizes of variants of $\mathsf{Sharp}_{\mathsf{GS}}$ in Bytes with correctness error $1 - p_{\mathsf{succ}}$. We give the proof size with the 3-square decomposition (π), the amortized proof size (π_{amor}), the proof size with the 4-square decomposition (π_{4sqr}), the proof size for the augmentation with an additional RSA group element (π_{RSA}) and the proof size for the augmentation with an additional class group element (π_{CL}).

λ	κ_{err}	B	Γ	L	N	p	q	R	p_{succ}	π	π_{amor}	π_{4sqr}	π_{RSA}	π_{CL}
128	40	32	41	10	1	256	256	1	0.993	234	234	244	667	455
128	40	32	41	10	4	256	256	1	0.982	358	90	400	792	579
128	40	32	41	10	8	256	256	1	0.966	524	66	607	958	745
128	40	32	41	10	16	256	256	1	0.937	856	54	1022	1290	1077
128	40	64	41	10	1	256	256	1	0.993	250	250	264	683	471
128	40	64	41	10	4	256	256	1	0.982	422	106	480	856	643
128	40	64	41	10	8	256	256	1	0.966	652	82	767	1086	873
128	40	64	41	10	16	256	256	1	0.937	1112	70	1342	1546	1333
128	80	32	81	10	1	256	256	1	0.993	254	254	269	687	475
128	80	32	81	10	4	256	256	1	0.982	438	110	500	872	659
128	80	32	81	10	8	256	256	1	0.966	684	86	807	1118	905
128	80	32	81	10	16	256	256	1	0.937	1176	74	1422	1610	1397
128	80	64	81	10	1	256	315	1	0.993	285	285	304	718	505
128	80	64	81	10	4	256	315	1	0.982	517	130	595	950	738
128	80	64	81	10	8	256	315	1	0.966	827	104	982	1260	1048
128	80	64	81	10	16	256	315	1	0.937	1447	91	1757	1880	1668
128	128	32	129	10	1	301	347	1	0.993	318	318	339	751	538
128	128	32	129	10	4	301	347	1	0.982	574	144	660	1007	795
128	128	32	129	10	8	301	347	1	0.966	916	115	1087	1349	1137
128	128	32	129	10	16	301	347	1	0.937	1600	100	1942	2033	1821
128	128	64	129	10	1	333	411	1	0.993	360	360	385	793	580
128	128	64	129	10	4	333	411	1	0.982	664	166	766	1097	885
128	128	64	129	10	8	333	411	1	0.966	1070	134	1273	1503	1291
128	128	64	129	10	16	333	411	1	0.937	1882	118	2288	2315	2103

Table 4: Overview of parameters (where $S = 2^{256} - 1$ always) and proof sizes of variants of $\mathsf{Sharp}^{\mathsf{Po}}_{\mathsf{SO}}$ in Bytes with correctness error $1 - p_{\mathsf{succ}}$. We give the proof size with the 3-square decomposition (π), the amortized proof size (π_{amor}), the proof size with the 4-square decomposition (π_{4sqr}), the proof size for the augmentation with an additional RSA group element (π_{RSA}) and the proof size for the augmentation with an additional class group element (π_{CL}).

λ	κ_{err}	B	Γ	L	N	p	R	p_{succ}	π	π_{amor}	π_{RSA}	π_{CL}
128	40	32	43	10	1	256	1	0.991	300	300	734	521
128	40	32	43	10	4	256	1	0.980	556	139	989	777
128	40	32	43	10	8	256	1	0.964	896	112	1329	1117
128	40	32	43	10	16	256	1	0.935	1576	99	2010	1797
128	40	64	43	10	1	256	1	0.991	324	324	758	545
128	40	64	43	10	4	256	1	0.980	628	157	1061	849
128	40	64	43	10	8	256	1	0.964	1032	129	1465	1253
128	40	64	43	10	16	256	1	0.935	1840	115	2274	2061
128	80	32	43	10	1	256	2	0.989	323	323	757	544
128	80	32	43	10	4	256	2	0.978	579	145	1013	800
128	80	32	43	10	8	256	2	0.963	920	115	1353	1141
128	80	32	43	10	16	256	2	0.933	1600	100	2034	1821
128	80	64	43	10	1	256	2	0.989	355	355	789	576
128	80	64	43	10	4	256	2	0.978	659	165	1093	880
128	80	64	43	10	8	256	2	0.963	1064	133	1497	1285
128	80	64	43	10	16	256	2	0.933	1872	117	2306	2093
128	128	32	67	10	1	256	2	0.989	335	335	769	556
128	128	32	67	10	4	256	2	0.978	591	148	1025	812
128	128	32	67	10	8	256	2	0.963	932	117	1365	1153
128	128	32	67	10	16	256	2	0.933	1612	101	2046	1833
128	129	64	46	10	1	256	3	0.987	389	389	822	609
128	128	64	35	10	4	256	4	0.974	714	179	1148	935
128	128	64	35	10	8	256	4	0.959	1119	140	1553	1340
128	128	64	35	10	16	256	4	0.929	1928	121	2362	2149
128	40	32	1	10	1	256	40	0.919	777	777	–	–
128	40	32	1	10	16	256	40	0.866	2092	131	–	–
128	40	64	1	10	1	256	40	0.919	1113	1113	–	–
128	40	64	1	10	16	256	40	0.866	2668	167	–	–
128	128	32	1	10	1	256	128	0.773	1877	1877	–	–
128	128	32	1	10	16	256	128	0.729	3280	205	–	–
128	128	64	1	10	1	256	128	0.773	2917	2917	–	–
128	128	64	1	10	16	256	128	0.729	4560	285	–	–