Sharp: Short Relaxed Range Proofs

Geoffroy Couteau, Dahmun Goudarzi, Michael Klooß, Michael Reichle

Sharp: Short Relaxed Range Proofs

Geoffroy Couteau
couteau@irif.fr
CNRS, IRIF, Université Paris Cité
Paris, France

Dahmun Goudarzi
dahmun.goudarzi@gmail.com
Unaffiliated
Paris, France

Michael Klooß
michael.klooss@kit.edu
Karlsruhe Institute of Technology, KASTEL
Karlsruhe, Germany

Michael Reichle
michael.reichle@ens.fr
DIENS, École normale supérieure, PSL University, CNRS, INRIA
Paris, France

ABSTRACT

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.

CCS CONCEPTS

• **Security and privacy** → *Cryptography*; • **Theory of computation** → **Interactive proof systems**; *Cryptographic protocols*; **Communication complexity**.

KEYWORDS

relaxed range proof, zero-knowledge, proof of knowledge, square decomposition, proof of shortness,

ACM Reference Format:
Geoffroy Couteau, Dahmun Goudarzi, Michael Klooß, and Michael Reichle. 2022. Sharp: Short Relaxed Range Proofs. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3548606.3560628

CCS '22, November 7–11, 2022, Los Angeles, CA, USA

ACM ISBN 978-1-4503-9450-5/22/11.
https://doi.org/10.1145/3548606.3560628

1 INTRODUCTION

Zero-Knowledge Proofs and Range Proofs. Zero-knowledge proofs, introduced in the seminal work of Goldwasser, Micali, and Rackoff [30], 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 [19], e-voting [31], or e-cash [15], and have been introduced recently in some popular anonymous cryptocurrencies (see [12, 27, 42]).

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 [14, 32], 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 [13], 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 [10, 23, 31, 36], where one proves a statement of the form $x \geq 0$ by using special *integer commitment schemes* [25, 29] 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 [22], 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 [23] 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 [43] or ristretto255 [26]). 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 [25]). 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 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

[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.

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 [22] 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 [22] (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 [22] 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 [10, 23, 31, 36]. 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 $\mathsf{Sharp}^{\mathrm{Po}}_{\mathrm{SO}}$, 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 [23]: 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 $\mathsf{Sharp}^{\mathrm{Po}}_{\mathrm{SO}}$, using the 256-bit elliptic curve from the libsecp256k1 library [43]. 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.

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 [22] 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 $\mathsf{Sharp}_{\mathrm{GS}}$*: 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 $\mathsf{Ped.Commit}(f, t_*) = D_* + \gamma C_*$. 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

[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.

Table 1: Theoretical proof size in Bytes for showing that some $x \in [0, B]$ of CKLR proofs [22], Bulletproofs [13], RSA-based range proofs [23] and Sharp proofs (Sharp_{GS}, $\text{Sharp}^{\text{Po}}_{\text{SO}}$ and $\text{Sharp}_{\text{RSA}}$) given the security parameter λ. The groups $\mathbb{G}_{\text{com}}$ and $\mathbb{G}_{\text{3sq}}$ 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	Sharp_{GS}			$\text{Sharp}^{\text{Po}}_{\text{SO}}$			$\text{Sharp}_{\text{RSA}}$
$(\lambda, \log B)$	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 [13], using batch sizes $N = 1$ and $N = 8$. Both implementations use the library libsecp256k1 [43], and were run on a MacBook Pro with a 2.3 GHz Intel core i7 processor. All timings are in milliseconds.

		Bulletproofs		$\text{Sharp}^{\text{Po}}_{\text{SO}}$	
$(\lambda, \log B)$	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

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 [22, 23].

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 $\text{Sharp}^{\text{Po}}_{\text{SO}}$*: Cheaper Repetitions via a Novel PoSO.* To clarify the requirements for our PoSO, we take a closer look at the security proof of Sharp_{GS}. The PoDec proves (among other equations) the square decomposition of N integers x_i:

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

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) 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) 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 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

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

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 theorem (theorem 3.3) 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 [5]. 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 [6, 7]. 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 [2].

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. [38, 39], though still an order of magnitude larger than group-based proofs, e.g. [38] 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 [10, 22, 23, 31, 36]. 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 $A+b := A+\{b\}$ 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 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 $|x| = \min\{|k| \mid k \in \mathbb{Z}, k \equiv_p x\} \leq p/2$.

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. $\mathsf{priml}(k) := \prod_{i=1}^k p_i$ 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 Cryptographic Primitives

We define syntax and semantics of cryptographic primitives, and sketch their security properties. For formal definitions, see the full version [21].

2.2.1 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.

[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, $\mathsf{priml}(k) = p_k\#$.

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 order, the DLOG assumption can be strengthened.

*Definition 2.2 (*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]

2.2.2 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 $\mathsf{Hash}(m) = \mathsf{Hash}(m')$.

2.2.3 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 (Setup, Commit, Verify) s.t.

- $\mathsf{Com.Setup}(1^\lambda)$: outputs a commitment key ck (often left implicit),
- $\mathsf{Com.Commit}_{\mathsf{ck}}(x)$: outputs a pair (c, d) of commitment c (to x) and opening d under commitment key ck,
- $\mathsf{Com.Verify}_{\mathsf{ck}}(c, x, d)$: 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 [40], 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 $\mathsf{ck} = (\{G_i\}_{i \in [0,N]})$. 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]

2.2.4 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.

2.2.5 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 [37]. 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.

2.3 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.

[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 [1], 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 [20, Section 3.2].

[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.

Definition 2.3. 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 2.4 (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).*

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

Remark 2.1. *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 [28].*

2.4 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 [3]. 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*, so that $\sup_{v \in [0,V]} \Pr[\mathsf{mask}(v, r) = \perp \mid r \xleftarrow{\$} \mathsf{R}] \leq \mathsf{p}$.
- 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]

[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 $[V, (V + 1)L]$.

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 $\mathsf{mask}(v, r) = v + r \mod p$ (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. 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 $\exists d \in [1, D] : d\vec{x} \in [-K, K]_{\mathbb{Z}_p}^N$. 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

$$\forall \vec{x} \in \mathbb{Z}_p^N : \phi_{K,D}(\vec{x}) = 0 \implies \Pr[T(\vec{x}) = 1] \leq \kappa. \tag{2}$$

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 $d \cdot \vec{x} \in [-K, K]_{\mathbb{Z}_p}^N$, 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 technical, and we refer to the full version [21] for details.

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 $\beta = \min(N, \mathrm{primlmin}(D+1))$ and suppose that $2D(K' + DK + 2) < p$. Then* RAST *is fractionally (K', D)-sound with error $8/(D+1)$,*

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 mask_x, mask_r, masking randomness distribution R_x, R_r, masking overhead L_x, L_r and masking abort probability p_x, p_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 $\mathsf{ck}_{\mathbb{G}_{\mathsf{com}}}$ and $H_0, H_i \xleftarrow{\$} \mathbb{G}_{\mathsf{3sq}}$ for $\mathsf{ck}_{\mathbb{G}_{\mathsf{3sq}}}$, 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 $\mathsf{crs} = (\mathsf{ck}_{\mathbb{G}_{\mathsf{com}}}, \mathsf{ck}_{\mathbb{G}_{\mathsf{3sq}}})$.

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 $\mathcal{D} = (D_{k,x}, D_{k,y}, D_{k,*})_{k=1}^{R}$ 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 $[(1 - \mathsf{p}_r)^3 \cdot (1 - \mathsf{p}_x)^{4N}]^R$.

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* $1 - [(1 - \mathsf{p}_r)^3 \cdot (1 - \mathsf{p}_x)^N]^R$. *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* $\mathsf{R}_{\mathsf{Ext}} = \{((x_i)_{i=1}^N, r_x) : C_x = r_x G_0 + \sum_{i=1}^N x_i G_i \wedge [x_i]_{\mathbb{Q}} \in [-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}_{K,\Gamma}}\}$. *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* $d \in [1, \Gamma]$.

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

Prover$(C_x, B, r_x, \{x_i\}_{i=1}^N)$		Verifier(C_x, B)
1: Compute $y_{i,j}$ s.t. $4x_i(B - x_i) + 1 = \sum_{j=1}^3 y_{i,j}^2$ for $i \in [1, N]$		
2: Set $C_y = r_y G_0 + \sum_{i=1}^N \sum_{j=1}^3 y_{i,j} G_{i,j}$ for $r_y \xleftarrow{\$} [0, S]$		
3: **for all** $k \in [1, R]$ **do**		
4: Set $\widetilde{r}_{k,x}, \widetilde{r}_{k,y} \xleftarrow{\$} \mathsf{R}_r$	▷ Opening	
5: Set $\widetilde{x}_{k,i}, \widetilde{y}_{k,i,j} \xleftarrow{\$} \mathsf{R}_x$ for $i \in [1, N]$, $j \in [1, 3]$		
6: Set $D_{k,x} = \widetilde{r}_{k,x} G_0 + \sum_{i=1}^N \widetilde{x}_{k,i} G_i$		
7: Set $D_{k,y} = \widetilde{r}_{k,y} G_0 + \sum_{i=1}^N \sum_{j=1}^3 \widetilde{y}_{k,i,j} G_{i,j}$		
8: Set $r_k^* \xleftarrow{\$} [0, S]$ and $\widetilde{r}_k^* \xleftarrow{\$} \mathsf{R}_r$	▷ Decomposition	
9: Set $\alpha_{1,k,i}^* = 4\widetilde{x}_{k,i} B - 8 x_i \widetilde{x}_{k,i} - 2 \sum_{j \in [1,3]} y_{i,j} \widetilde{y}_{k,i,j}$ for $i \in [1, N]$		
10: Set $\alpha_{0,k,i}^* = -(4\widetilde{x}_{k,i}^2 + \sum_{j \in [1,3]} \widetilde{y}_{k,i,j}^2)$ for $i \in [1, N]$		
11: Set $C_{k,*} = r_k^* H_0 + \sum_{i=1}^N \alpha_{1,k,i}^* H_i$		
12: Set $D_{k,*} = \widetilde{r}_k^* H_0 + \sum_{i=1}^N \alpha_{0,k,i}^* H_i$		
	$C_y, \{C_{k,*}, D_{k,x}, D_{k,y}, D_{k,*}\}_{k=1}^R$ →	
		1: $\gamma_k \xleftarrow{\$} [0, \Gamma]$ for $k \in [1, R]$ ▷ Challenge
	← $\{\gamma_k\}_{k=1}^R$	
13: **for all** $k \in [1, R]$, $i \in [1, N]$, $j \in [1, 3]$ **do**		
14: Set $z_{k,i} = \mathsf{mask}_x(\gamma_k \cdot x_i, \widetilde{x}_{k,i})$, $z_{k,i,j} = \mathsf{mask}_x(\gamma_k \cdot y_{i,j}, \widetilde{y}_{k,i,j})$		
15: Set $t_{k,x} = \mathsf{mask}_r(\gamma_k r_x, \widetilde{r}_{k,x})$, $t_{k,y} = \mathsf{mask}_r(\gamma_k \cdot r_y, \widetilde{r}_{k,y})$		
16: Set $t_k^* = \mathsf{mask}_r(\gamma_k \cdot r_k^*, \widetilde{r}_k^*)$		
17: **if** any $z_{k,i}$, $t_{k,x}$ or t_k^* is $\perp$ **then**		
18: **abort**	▷ Masking failed	
	$\{z_{k,i,j}, z_{k,i}, t_{k,x}, t_{k,y}, t_k^*\}_{k \in [1,R], i \in [1,N], j \in [1,3]}$ →	
		2: **for all** $k \in [1, R]$ **do**
		3: Check $D_{k,x} + \gamma_k C_x = t_{k,x} G_0 + \sum_{i=1}^N z_{k,i} G_i$
		4: Check $D_{k,y} + \gamma_k C_y = t_{k,y} G_0 + \sum_{i=1}^N \sum_{j=1}^3 z_{k,i,j} G_{i,j}$
		5: Set $f_{k,i}^* = 4 z_{k,i}(\gamma_k B - z_{k,i}) + \gamma_k^2 - \sum_{j=1}^3 z_{k,i,j}^2$
		6: Check $D_{k,*} + \gamma_k C_{k,*} = t_k^* H_0 + \sum_{i=1}^N f_{k,i}^* H_i$
		7: Check $z_{k,i}, z_{k,i,j} \in [0, (B\Gamma + 1)L_x]$ for $i \in [1, N]$, $j \in [1, 3]$
		8: **return** 1 iff all checks succeed

Security proof, outline. Here, we only sketch the proof of security and the relaxed soundness guarantee. We refer to the full version [21] 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 2.4). 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 $[x_i]_{\mathbb{Q}} \in [-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}}$. 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 $\mathsf{ck}_{\mathsf{com}}$ is augmented by additional elements $\widetilde{G}_j \xleftarrow{\$} \mathbb{G}_{\mathsf{com}}$ for $j \in [1, R]$. For simplicity, we define $\widehat{\Gamma} := (\Gamma+1)^R - 1$ (the size of "large" challenge), and require that $\widehat{\Gamma} \le p$.[11]

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 $V_\mu = \mathsf{R}_{poso} \cdot \widehat{\Gamma} L$. 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 $x_i \in [0, B_i]$.

[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.

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.

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

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

5.3 Security and Correctness

Non-abort probability. With R "internal" repetitions, the non-abort probability 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* $2(\Gamma+1)^2 K' < p$ *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* $\mathsf{R}_{\mathsf{Ext}} = \{((x_i)_{i=1}^N, r_x) : C_x = r_x G_0 + \sum_{i=1}^N x_i G_i \wedge [x_i]_{\mathbb{Q}} \in [-\frac{1}{4B}, B + \frac{1}{4B}]_{\mathbb{Q}_{K',\Gamma}}\}$. *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* $d \in [1, \Gamma]$.

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 $D_* = \widetilde{r}^* H_0 + \sum_{i=1}^{N} \alpha^*_{0,i} H_i$

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

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

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 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. Also, since Phase 2 is effectively independent of Phase 1, it may be exchanged with other suitable (succinct) argument systems.

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.

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 $\kappa_{\mathsf{err}} = 2^{-40} + \mathsf{negl}$ 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).

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 the full version [21].

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

[12] In this case, the communication overhead is reasonable and computational efficiency remains excellent. For 128 repetitions, the communication overhead becomes noticeable. See the full version [21] 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.

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 $\mathsf{Sharp}_{\mathrm{GS}}$ and $\mathsf{Sharp}^{\mathrm{Po}}_{\mathrm{SO}}$, 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).

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 [22].

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 [11, 16, 19] 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 [17]. 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 $\mathsf{Sharp}_{\mathrm{RSA}}$ 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 [18], 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 [4, 24, 41], 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 [14] allows quite efficient digit-based decompositions. However, operations in pairing-groups are slower, elements are bigger, and for efficiency, [14] needs relatively large (non-transparent) public parameters.

7.2 Updatable Anonymous Credentials and BBAs

A line of works [8, 9, 33–35] 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 [33, 35] consider weakened (security) requirements to achieve practical efficiency. Even in later works [8, 9, 34], 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 [12, 27, 42]. This ensures that no user can spend more coins than he owns while preserving privacy. Thus, this is a "secret balance update" setting. Here, we give an overview on the applicability of Sharp in this context and refer to the full version [21] for more details.

When a sender with a balance of b coins performs a transfer of a coins to a receiver, she has to guarantee the following: (1) $b - a \geq 0$, i.e. the sender's balance remains non-negative after the transaction and (2) $a \geq 0$, i.e. the sender transfers a non-negative number of coins to the receiver. Often, the values a and b are committed (or fixed via an encryption), and the sender performs two range proofs to show equations (1) and (2). Unfortunately, even an initial shortness guarantee on the committed balances b is not sufficient for relaxed soundness to provide standard guarantees, as the shortness of a cannot be guaranteed this way. Thus, we cannot replace *all* range proofs with Sharp proofs naively (and doing so would lead to concrete attacks). Nevertheless, *some* range proofs can be replaced with Sharp proofs for efficiency improvements.

Furthermore, in the full version [21] we sketch how the use of augmented Sharp proofs, with both an additional RSA and class group element, is sufficient to avoid these attacks *without* trusted setup of the RSA modulus. Perhaps surprisingly, we can still leverage the properties of RSA groups in this case.

ACKNOWLEDGMENTS

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

[1] Damiano Abram, Ivan Damgård, Claudio Orlandi, and Peter Scholl. 2022. An Algebraic Framework for Silent Preprocessing with Trustless Setup and Active Security. In *Advances in Cryptology – CRYPTO 2022*, Yevgeniy Dodis and Thomas Shrimpton (Eds.). Springer Nature Switzerland.
[2] Martin R. Albrecht and Russell W. F. Lai. 2021. Subtractive Sets over Cyclotomic Rings - Limits of Schnorr-Like Arguments over Lattices. 519–548. https://doi.org/10.1007/978-3-030-84245-1_18
[3] Thomas Attema, Ronald Cramer, and Lisa Kohl. 2021. A Compressed Σ-Protocol Theory for Lattices. 549–579. https://doi.org/10.1007/978-3-030-84245-1_19
[4] Foteini Baldimtsi and Anna Lysyanskaya. 2013. Anonymous credentials light. 1087–1098. https://doi.org/10.1145/2508859.2516687
[5] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. 2018. Sub-linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits. 669–699. https://doi.org/10.1007/978-3-319-96881-0_23
[6] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. 2018. More Efficient Commitments from Structured Lattice Assumptions. 368–385. https://doi.org/10.1007/978-3-319-98113-0_20
[7] Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. 2015. Efficient Zero-Knowledge Proofs for Commitments from Learning with Errors over Rings. 305–325. https://doi.org/10.1007/978-3-319-24174-6_16
[8] Johannes Blömer, Jan Bobolz, Denis Diemert, and Fabian Eidens. 2019. Updatable Anonymous Credentials and Applications to Incentive Systems. 1671–1685. https://doi.org/10.1145/3319535.3354223
[9] Jan Bobolz, Fabian Eidens, Stephan Krenn, Daniel Slamanig, and Christoph Striecks. 2020. Privacy-Preserving Incentive Systems with Highly Efficient Point-Collection. 319–333. https://doi.org/10.1145/3320269.3384769
[10] Fabrice Boudot. 2000. Efficient Proofs that a Committed Number Lies in an Interval. 431–444. https://doi.org/10.1007/3-540-45539-6_31
[11] Stefan Brands. 2000. *Rethinking public key infrastructures and digital certificates: building in privacy*. MIT Press.
[12] Benedikt Bünz, Shashank Agrawal, Mahdi Zamani, and Dan Boneh. 2020. Zether: Towards Privacy in a Smart Contract World. 423–443. https://doi.org/10.1007/978-3-030-51280-4_23
[13] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. 2018. Bulletproofs: Short Proofs for Confidential Transactions and More. 315–334. https://doi.org/10.1109/SP.2018.00020
[14] Jan Camenisch, Rafik Chaabouni, and abhi shelat. 2008. Efficient Protocols for Set Membership and Range Proofs. 234–252. https://doi.org/10.1007/978-3-540-89255-7_15
[15] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. 2005. Compact E-Cash. 302–321. https://doi.org/10.1007/11426639_18
[16] Jan Camenisch and Anna Lysyanskaya. 2001. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. 93–118. https://doi.org/10.1007/3-540-44987-6_7
[17] Jan Camenisch and Anna Lysyanskaya. 2003. A Signature Scheme with Efficient Protocols. 268–289. https://doi.org/10.1007/3-540-36413-7_20
[18] Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. 2014. Algebraic MACs and Keyed-Verification Anonymous Credentials. 1205–1216. https://doi.org/10.1145/2660267.2660328
[19] David Chaum. 1990. Showing Credentials without Identification Transferring Signatures between Unconditionally Unlinkable Pseudonyms. 246–264. https://doi.org/10.1007/BFb0030366
[20] Henry Corrigan-Gibbs and Dmitry Kogan. 2018. The Discrete-Logarithm Problem with Preprocessing. 415–447. https://doi.org/10.1007/978-3-319-78375-8_14

[21] Geoffroy Couteau, Dahmun Goudarzi, Michael Klooß, and Michael Reichle. 2022. Sharp: Short Relaxed Range Proofs. Cryptology ePrint Archive, Paper 2022/1153. https://doi.org/10.1145/3548606.3560628 https://eprint.iacr.org/2022/1153.
[22] Geoffroy Couteau, Michael Klooß, Huang Lin, and Michael Reichle. 2021. Efficient Range Proofs with Transparent Setup from Bounded Integer Commitments. 247–277. https://doi.org/10.1007/978-3-030-77883-5_9
[23] Geoffroy Couteau, Thomas Peters, and David Pointcheval. 2017. Removing the Strong RSA Assumption from Arguments over the Integers. 321–350. https://doi.org/10.1007/978-3-319-56614-6_11
[24] Geoffroy Couteau and Michael Reichle. 2019. Non-interactive Keyed-Verification Anonymous Credentials. 66–96. https://doi.org/10.1007/978-3-030-17253-4_3
[25] Ivan Damgård and Eiichiro Fujisaki. 2002. A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order. 125–142. https://doi.org/10.1007/3-540-36178-2_8
[26] Henry de Valence, Jack Grigg, George Tankersley, Filippo Valsorda, and Isis Lovecruft. 2019. *The ristretto255 group.* Technical Report. IETF CFRG Internet Draft.
[27] The Zcash developers. 2022. Zcash. https://github.com/zcash/zcash.
[28] Pierre-Alain Fouque, Jacques Stern, and Jan-Geert Wackers. 2003. CryptoComputing with Rationals. 136–146.
[29] Eiichiro Fujisaki and Tatsuaki Okamoto. 1997. Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations. 16–30. https://doi.org/10.1007/BFb0052225
[30] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. 1989. The Knowledge Complexity of Interactive Proof Systems. 18, 1 (1989), 186–208.
[31] Jens Groth. 2005. Non-interactive Zero-Knowledge Arguments for Voting. 467–482. https://doi.org/10.1007/11496137_32
[32] Jens Groth. 2011. Efficient Zero-Knowledge Arguments from Two-Tiered Homomorphic Commitments. 431–448. https://doi.org/10.1007/978-3-642-25385-0_23
[33] Gunnar Hartung, Max Hoffmann, Matthias Nagel, and Andy Rupp. 2017. BBA+: Improving the Security and Applicability of Privacy-Preserving Point Collection. 1925–1942. https://doi.org/10.1145/3133956.3134071
[34] Max Hoffmann, Michael Klooß, Markus Raiber, and Andy Rupp. 2020. Black-Box Wallets: Fast Anonymous Two-Way Payments for Constrained Devices. *Proc. Priv. Enhancing Technol.* 2020, 1 (2020), 165–194. https://doi.org/10.2478/popets-2020-0010
[35] Tibor Jager and Andy Rupp. 2016. Black-Box Accumulation: Collecting Incentives in a Privacy-Preserving Way. *Proc. Priv. Enhancing Technol.* 2016, 3 (2016), 62–82. https://doi.org/10.1515/popets-2016-0016
[36] Helger Lipmaa. 2003. On Diophantine Complexity and Statistical Zero-Knowledge Arguments. 398–415. https://doi.org/10.1007/978-3-540-40061-5_26
[37] Vadim Lyubashevsky. 2009. Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures. 598–616. https://doi.org/10.1007/978-3-642-10366-7_35
[38] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plancon. 2022. Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General. In *Advances in Cryptology – CRYPTO 2022*, Yevgeniy Dodis and Thomas Shrimpton (Eds.). Springer Nature Switzerland.
[39] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. 2020. Practical Lattice-Based Zero-Knowledge Proofs for Integer Relations. 1051–1070. https://doi.org/10.1145/3372297.3417894
[40] Torben P. Pedersen. 1992. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. 129–140. https://doi.org/10.1007/3-540-46766-1_9
[41] Sietse Ringers, Eric R. Verheul, and Jaap-Henk Hoepman. 2017. An Efficient Self-blindable Attribute-Based Credential Scheme. 3–20.
[42] The Monero Project. 2022. Monero. https://github.com/monero-project/monero.
[43] Pieter Wuille. 2018. libsecp256k1. https://github.com/bitcoin/secp256k1.