Efficient Range Proofs with Transparent Setup from Bounded Integer Commitments

Geoffroy Couteau[1], Michael Klooß[2], Huang Lin[3], and Michael Reichle[4,5]

[1] CNRS, IRIF, Université de Paris, France, couteau@irif.fr
[2] Karlsruhe Institute for Technology, michael.klooss@kit.edu
[3] Mercury's Wing and Suterusu Project, huanglinepfl@gmail.com
[4] DIENS, École normale supérieure, CNRS, PSL University, 75005 Paris, France
[5] Inria, Paris, France, michael.reichle@ens.fr

Abstract. We introduce a new approach for constructing range proofs. Our approach is modular, and leads to highly competitive range proofs under standard assumption, using less communication and (much) less computation than the state of the art methods, and without relying on a trusted setup. Our range proofs can be used as a drop-in replacement in a variety of protocols such as distributed ledgers, anonymous transaction systems, and many more, leading to significant reductions in communication and computation for these applications.

At the heart of our result is a new method to transform any commitment over a finite field into a commitment scheme which allows to commit to and efficiently prove relations about *bounded integers*. Combining these new commitments with a classical approach for range proofs based on square decomposition, we obtain several new instantiations of a paradigm which was previously limited to RSA-based range proofs (with high communication and computation, and trusted setup). More specifically, we get:

- Under the discrete logarithm assumption, we obtain the most compact and efficient range proof among all existing candidates (with or without trusted setup). Our proofs are 12% to 20% shorter than the state of the art Bulletproof (Bootle et al., CRYPTO'18) for standard choices of range size and security parameter, and are more efficient (both for the prover and the verifier) by more than an order of magnitude.
- Under the LWE assumption, we obtain range proofs that improve over the state of the art in a batch setting when at least a few dozen range proofs are required. The amortized communication of our range proofs improves by up to two orders of magnitudes over the state of the art when the number of required range proofs grows.
- Eventually, under standard class group assumptions, we obtain the first concretely efficient standard integer commitment scheme (without bounds on the size of the committed integer) which does not assume trusted setup.

Keywords. Range Proof, Integer Commitments.

1 Introduction

In this work, we develop new techniques to construct range proofs, an important building block in a variety of modern cryptographic protocols such as distributed ledgers, anonymous transactions, e-cash, e-voting, and many more. The range proofs obtained with our methods are highly competitive with the state of the art: they rely on standard assumptions, require less communication and computation, and do not assume any trusted setup. Furthermore, our approach is modular and can be instantiated in the discrete logarithm setting, in the lattice setting (leading to the most efficient post-quantum range proofs in a batch setting), and in the class group setting. Below, we review some background.

Range proofs and anonymous transactions. Zero-knowledge proofs, introduced in the seminal work of Goldwasser, Micali, and Rackoff [GMR89], allow a prover to convince a verifier that a statement is true, while concealing all information beyond the truth of the statement. They are a fundamental primitive in cryptography, with inumerable applications. Range proofs, whose genesis can be traced back to [BCDv88], are a particular type of zero-knowledge proof where the prover wishes to convince the verifier that a committed value belongs to a certain range. Range proofs are a core building block in numerous applications such as anonymous credentials [Cha90], e-voting [Gro05], and e-cash [CHL05]. Furthermore, efficient range proofs have recently become central components in distributed ledgers, the prime example being the recent integration of Bulletproof [BBB$^+$18] in the cryptocurrency Monero[6] and later Mimblewimble-based anonymous cryptocurrencies such as Beam[7] and Grin[8]. Range proofs also play an essential role in anonymous payment schemes for smart contract platforms such as Zether [BAZB20].

In most of these anonymous payment schemes, (positive and negative) integers are encoded as finite field elements, and negative spendings constitute a valid transaction in general, if they are not explicitly disallowed. This feature can be exploited to launch a double-spending attack, allowing the adversary to print money out of thin air [MIO18]. In a confidential payment scheme where both inputs and outputs of a transaction are hidden in either a digital commitment (as in Monero) or an encryption (as in Zether), range proofs are necessary to guarantee that the hidden value falls into the correct range and prevent the aforementioned overflow attack.

The maximum throughput of a distributed ledger protocol is mainly determined by the maximum block size and average transaction size [CDE$^+$16]. The smaller the transaction size is, the larger the maximum throughput is. The average transaction size in an anonymous payment scheme is largely determined by the zero-knowledge range proof size. Therefore, the proof size is a crucial parameter

[6] `https://web.getmonero.org/resources/moneropedia/bulletproofs.html`
[7] `https://github.com/BeamMW/beam`
[8] `https://cointelegraph.com/news/cryptocurrency-grin-follows-through-with-anticipated-july-17-mainnet-hardfork`

for the design of a range proof scheme. The proof generation and verification time are also vital to the performance of the system built on the range proof scheme. In the case of a decentralized anonymous payment scheme, the proof generation time will determine how fast the anonymous payment can be launched and have a direct impact on the user experience and system scalability [CZJ+17]. The proof verification time, on the other hand, has a great impact on the workload of the miners.

1.1 Standard Approaches for Building Range Proofs

Due to their wide variety of applications, many constructions of range proofs have been proposed over the past decades. All these constructions can be categorized in two main high level approaches, which we outline below.

First method: n-ary decomposition. The first method is the one employed both in the early (folklore) constructions of range proofs, as well as in the latest state-of-the-art constructions (such as Bulletproof). To prove that a committed integer x belong to an interval of the form $[0, n^\ell - 1]$, where n is some small value, this method uses the following high-level template:

1. First, commit to the n-ary decomposition of x, denoted $(x_0, \cdots, x_{\ell-1})$.
2. Second, prove that the relation $x = \sum_{i=0}^{\ell-1} x_i \cdot n^i$ holds.
3. Third, prove that each component of the committed tuple belongs to $[0, n-1]$. Since n is typically very small, this can be achieved using some brute-force method (for example, when using binary decomposition, it amounts to proving that each component is a bit, which can be done using standard methods).

When the commitment scheme satisfies some homomorphic properties, it is generally simple to lift a proof as above to a proof for a more general interval $[a, b]$. The first instance of this approach is a folklore discrete-logarithm-based construction using the Pedersen commitment scheme to commit to the bit decomposition of x. Denoting $\beta = \log(b-a)$ the bitlength of the interval size and λ the bitlength of group elements, This leads to a range proof communicating $\mathcal{O}(\lambda \cdot \beta)$ bits. This approach was first improved in [CCs08] to $\mathcal{O}(\lambda \cdot \beta / \log \beta)$ by using decomposition in a larger basis, and later in [Gro11] to $\mathcal{O}(\lambda \cdot \beta^{1/3})$, using pairings.

In a recent breakthrough work, the authors of [BBB+18] introduced Bulletproof, which managed to reduce the communication to $\mathcal{O}(\lambda \cdot \log \beta)$ under the plain DLOG assumption (without pairings) while still remaining computationally efficient. Their approach relies on generalized Pedersen commitment to commit to the entire bit-decomposition of x using few group elements, and on a clever recursive proof strategy to simultaneously prove that all committed values are bits.[9] This comes at the cost of a larger number of rounds $\mathcal{O}(\log \beta)$ (but this is

[9] There have been several recent follow up works [HKR19,AC20] to Bulletproof, which expand the set of relations captured by the framework, but do not translate into concrete improvements on the size of the range proofs produced by this framework.

typically not a concern in real-world applications, where the Fiat-Shamir heuristic is used to make the proof non-interactive) and a computational soundness guarantee (leading to a zero-knowledge argument instead of a proof).

A strong advantage of the proofs obtained in this line of work is that they do not require any trusted setup. In real-world applications such as cryptocurrencies, this is an important feature to avoid having to trust any central authority with the secure generation of the parameters (we will discuss this more later). Due to this feature and its good concrete efficiency, Bulletproof is currently considered the state of the art method for range proofs, and has found its way into several real-world protocols.

Second method: square decomposition. The second method can be traced back to the work of Boudot [Bou00], and was initially introduced to avoid the large $\mathcal{O}(\lambda \cdot \beta)$ cost of the range proofs obtained (at the time) by the first method. It relies on the following high-level template (or a close variant thereof): first, proving that $x \in [a, b]$ reduces to proving that $x - a$ and $b - x$ (whose commitments can typically be computed homomorphically from a commitment to x) are positive. Now, to prove that a committed value y is positive:

1. First, decompose y as $y = \sum_{i=1}^{4} y_i^2$ over the integers. Lagrange's four square theorem guarantees that such a decomposition exists, and efficient algorithms allow to quickly find one.
2. Second, commit to the y_i and prove (using standard methods) that $y = \sum_{i=1}^{4} y_i^2$ *over the integers*.

The advantage of this method is that it requires committing only to a constant number of components (independent of the interval size), instead of $\approx \beta$ components with the first method. This typically leads to proofs with communication $\mathcal{O}(\beta + \lambda)$ bits. However, it is crucial for this method that the relation is proven *over the integers*: standard commitment schemes such as Pedersen only allow committing values over $\mathbb{Z}_p$ for some prime p, but finding a 4-square decomposition over $\mathbb{Z}_p$ does not provide any guarantee of positivity. Hence, a core component of this line of work is the notion of *integer commitment schemes*, introduced in [FO97,DF02], which allows to commit and prove relations among values directly over the integers.

The square decomposition method has been refined in [Lip03]. Later, the work of Groth [Gro05] observed that one can instead decompose $4y + 1$ as a sum of three squares (positive integers congruent to 1 modulo 4 can always be decomposed this way) to reduce the proof size, and further efficiency and security improvements were described in [CPP17]. A common issue of all these works is that all known integer commitment schemes require the use of RSA groups or class groups with a hard-to-factor discriminant. This means that the group size is very large (typically 3072 bits), and that these proofs all require a trusted setup to generate a public product of secret prime factors[10]. Assuming

[10] While it is theoretically possible to use a very large random integer as RSA modulus, without relying on a trusted party to compute a product of safe primes, this approach

a trusted setup is a rather undesirable property in a decentralized anonymous payment scheme: in general, the party responsible for the setup step can exploit the trapdoor information obtained through this process to print an unlimited amount of cryptocurrency without being detected [Sle,Ben]. Although one could potentially mitigate the risk of the above attack by using secure multi-party computation to execute the setup step (as was done e.g. for zcash[11]), it introduces additional engineering complexity and potential vulnerabilities.

Furthermore, even before Bulletproof, these proof systems were competitive with proofs obtained with the first method only for very large intervals. Compared to Bulletproof, they lead to much larger proof sizes for any interval size (and are also computationally less efficient). Due to their higher cost and their need of a trusted setup, this second method is largely considered obsolete and non-competitive with the proofs obtained through the first method.

1.2 Our Contribution

In this work, we turn the tables and demonstrate that the square decomposition method can be refined to create highly competitive range proofs, with smaller communication and computation compared to the state of the art Bulletproof, without trusted setup (meaning that our proofs only require a transparent setup), and under standard assumptions. Among other advantages, our method is modular and can also be instantiated in the lattice setting to obtain post-quantum range proofs which are highly competitive with the sate of the art in a batch scenario (where several range proofs must be computed at once), and in the class group setting with prime discriminant. Furthermore, our proofs require only three rounds of interaction, an important feature if one does not want to rely on the Fiat-Shamir heuristic, and can be modified to achieve statistical soundness instead of computational soundness (at a small cost in efficiency). At the heart of our constructions is a new generic method to convert any commitment scheme over $\mathbb{Z}_p$ into a *bounded integer commitment scheme*, i.e., a commitment scheme which allows to commit to bounded-range integers and to prove relations over $\mathbb{Z}$ between committed bounded-range integers.

Instantiation in the discrete-log setting. Instantiating our framework with the standard Pedersen commitment scheme, we obtain a bounded integer commitment scheme under the discrete logarithm assumption. When plugging this bounded integer commitment scheme in the range proof of [CPP17], we obtain a range proof which does not require any trusted setup and can benefit simultaneously from the compactness of square-decomposition-based range proofs (i.e., constant number of group elements) and the possibility of instantiating the Pedersen commitment scheme over prime-order elliptic curve, with small group

is completely impractical due to the very large group size and amount of computation, see the discussion on RSA-UFO in [LM19].

[11] `https://z.cash/technology/paramgen/`

elements[12]. To further optimize the proof size, we describe an optimized variant which relies on the *short-exponents* discrete logarithm assumption (i.e., the assumption that it is hard to compute discrete logarithm even when the exponent is sampled from a large enough bounded range), which is a well-studied variant of the standard discrete log assumption. For example, for an interval size of 2^{32} and 128 bits of security, we obtain range proofs of size 501 Bytes, compared to the 608 Bytes of Bulletproof. For the same parameters, the computational cost for both the prover and the verifier are more than an order of magnitude smaller compared to Bulletproof. The high efficiency of prover *and* verifier is crucial for use of (range) proofs on resource constrained devices, such as smartphones. Such devices are of special interest for privacy-enhancing technologies, such as anonymous credentials [Cha90] and payment systems. To achieve practicalility, tradeoffs have to made. For example, the work [BBDE19] relies on [CCs08], which requires pairings and relatively large public parameters, whereas the work [HKRR20] relies on *uncompressed*, i.e. linear-size, Bulletproofs, trading communication for computation. Our range proofs are a great fit for these settings.

Detailed comparison with Bulletproof. A more detailed comparison with Bulletproof is given in Table 1. Below, we explain how the numbers in the table have been obtained. Computing the exact costs of our range proof is rather tedious, since it involves careful optimizations with rejection sampling techniques, and optimizations using the short-exponent discrete logarithm assumption. We consider range proofs over an interval $[a, b]$ with $\beta = \log(b - a) \in \{32, 64\}$, a security parameter $\lambda \in \{80, 128\}$, and a group of size q (which might not be the same for Bulletproof and our range proof). The formula below additionally uses parameters C, S, L' corresponding respectively to the challenge size, a bound on the length of short exponents, and a bound for rejection sampling. Our concrete numbers are obtained by setting $C = 2^{\lambda}, S = 2^{2\lambda}, L' = \lceil 256\sqrt{2\lambda} \rceil$. The formulas for computing the range proof size (in the non-interactive setting, when Fiat-Shamir is used), the prover work, and the verifier work, are given below:

- Proof size (in bits): $30(\beta + \log(CL')) + \lceil \log(C)/\lambda \rceil (2\lambda + 4(2\beta + \log(CL') + 2\log(SCL')) + 2)$ (our work) versus $\log q \cdot (2\beta + 9)$ (Bulletproof).
- Prover work (in group multiplications): $2.31 \cdot (4\beta + 8\log S + 6\log C + 7\log L') + 30$ (our work) versus $18 \cdot (\beta \log q)$ (Bulletproof).
- Verifier work (in group multiplications): $4.5\beta + 7\log S + 13\log C + 9\log L' + 10$ versus at least $3\beta \cdot \log q$ (lower bound on the cost for Bulletproof, computed as the cost of a single inner product argument)
- Group size (in bits): $\log q = 32(2^{\beta} CL')^2 + 1$ (our work) versus $\log q = 2\lambda$ (Bulletproof)

[12] Since our bounded integer commitment scheme requires the committed values to remain into a bounded range, we actually require slightly larger group size compared to Bulletproof to achieve the same security level; this is accounted for in our concrete comparison and will be covered in details in the technical overview.

In the above, prover and verifier work are computed as the number of multiplications required for the exponentiations (we do not directly count the exponentiations for fairness of comparison: Bulletproof and our work do not use the same group size, and our optimized construction also uses exponentiations with short exponents), which largely dominate the overall cost. We note that in both our work and Bulletproof, the verifier work can be optimized by relying on multiexponentiations techniques; since these techniques apply identically in both works and do not significantly change the bottom line in terms of comparisons, we ignore them in this overview.

Asymptotically, our proofs have size $O(\lambda + \beta)$, while Bulletproof has size $O(\lambda \log \beta)$. We note that in the range of parameters $\beta = O(\lambda)$, our techniques actually leads to an asymptotic improvement over Bulletproof; for larger ranges, Bulletproof is more efficient, and for very small ranges, the asymptotic costs are the same for both. Previous square-decomposition-based range proofs had asymptotic cost $O(\beta + \lambda^{3-o(1)})$ due to their use of RSA modulus (which allow for subexponential attacks).

We stress that when not using the Fiat-Shamir heuristic, our scheme can be instantiated to have only three rounds (this slightly increases the proof size, because it requires to not use rejection sampling, since the latter causes the protocol to restart with non-negligible probability) while Bulletproof requires $\log \beta$ rounds. Even with rejection sampling and our concrete choice of parameters, the expected number of rounds is less than 5. Thus for sufficiently large β, our security proof is tighter than the one of Bulletproofs in the random-oracle model.

Furthermore, our scheme can be instantiated to have statistical soundness. On the other hand, Bulletproof allows for extremely efficient batching a large number of range proofs, and would therefore become preferable communication-wise when many range proofs must be performed at once. In any case, and independently of the number of range proofs, our range proofs requires 20 to 40 times less group multiplications for the prover, and 6 to 15 times less for the verifier.

Instantiation in the lattice setting. For the instantiation of our framework in the lattice setting, we build upon the commitment scheme and proof system from [YAZ+19]. The commitments built this way allow to commit to long vectors over $\mathbb{Z}_q^n$ (think of n as being a few thousands, e.g. $n = 5000$). Our techniques require to use a relatively large modulus q in order to avoid overflows in the computation. As a consequence, our commitments and proofs are quite large.

However, in exchange for using a large modulus, the commitment and proof system obtained by compiling the commitment of [YAZ+19] with our techniques allow to *batch* many range proofs extremely efficiently: we can essentially perform up to n range proofs in parallel for the cost of a single range proof, even if range proofs have different ranges. This improves over the communication achieved by the best LWE-based range proofs [YAZ+19]. Even compared to the more recent scheme of [BLLS20], which achieves very compact (single-shot) range proofs under the ring-SIS assumption, our approach starts to become more efficient from about 35 range proofs (and the efficiency gain scales linearly after that). In

Table 1. Comparison between the optimized range proof of Section 5.4 and Bulletproof [BBB+18] for various choices of security parameter λ and log of interval size β. Proof size and group size are in Bytes, prover and verifier work are counted as a number of group multiplications, rounded to two decimal places. See the paragraph "detailed comparison with Bulletproof" for the details on our computations.

(β, λ)		proof size	prover work	verifier work	Group size
$(32, 80)$	This Work	339	4.6k	2.4k	32
	Bulletproof	380	92k	$>$ 15k	20
$(32, 128)$	This Work	501	7k	3.7k	44
	Bulletproof	608	150k	$>$ 25k	32
$(64, 80)$	This Work	383	4.9k	2.6k	40
	Bulletproof	420	180k	$>$ 31k	20
$(64, 128)$	This Work	545	7.3k	3.8k	52
	Bulletproof	672	290k	$>$ 49k	32

the limit, when performing a large number of range proofs in parallel, we achieve about two orders of magnitude of communication reduction compared to the state of the art. The comparison is summarized on Table 2.

Table 2. Comparison of the range proof size in the lattice setting. Note that the scheme of [YAZ+19] was designed for large ranges. For a fair comparison, we apply similar vector-based batching optimization. The size is given in KB.

Range		batch size	$\lambda = 80$	$\lambda = 128$
$\beta = 32$	LWE [YAZ+19]	31	39	73
	This Work	35	4.7	5.2
	This Work	5000	0.1	0.1
$\beta = 64$	LWE [YAZ+19]	31	77	146
	This Work	35	8.4	9.7
	This Work	5000	0.16	0.16

Instantiation in the class group setting. Eventually, we also instantiate our method in the class group setting. The proofs obtained this way improve over our DLOG-based proofs only for large ranges, where Bulletproof would be more efficient. On the other hand, instantiating our approach in the class group setting leads to the first concretely efficient construction of *unbounded* integer commitment scheme which does not require a trusted setup (the only known alternative uses RSA-UFO, which is impractical, see the discussion in [LM19]).

Concurrent Works. In the DLOG setting, the work of [CHJ+20] recently claimed an improvement in proof size compared to [BBB+18] by slightly reducing the number of group elements required in [BBB+18]. The computational cost of their proof is the same as in [BBB+18]. To our knowledge, their scheme was not peer reviewed yet; we note that our range proofs are still shorter than theirs, and more than an order of magnitude computationally more efficient.

2 Technical Overview

As we outlined in the introduction, at the heart of our results is a method to convert standard homomorphic commitment schemes into *bounded integer commitment schemes* – that is, a scheme that allows to commit to integers from a bounded range, but also to *prove in zero-knowledge* relations between commited values *over the integers*, see [FO97,DF02] – with a certain set of additional specific properties. We now provide details on our approach.

2.1 A Natural Approach via Σ-Protocols

For simplicity, suppose that we have at our disposal a commitment scheme com with message space and random coin space $\mathbb{Z}_q$, for some large prime q, which is homomorphic over the messages and the coins: $\mathsf{com}(m_1; r_1) \cdot \mathsf{com}(m_2; r_2) = \mathsf{com}(m_1 + m_2; r_1 + r_2)$. This is satisfied for example by the Pedersen commitment scheme $\mathsf{com}(m; r) = g^m h^r$ for two group elements (g, h) over a group of order q. The transformation works for a more general class of commitments, this choice of structure is for the sake of concreteness. Suppose now that we would like to obtain a bounded integer commitment scheme out of com. The first obvious idea is to proceed as follows:

- map values in $\mathbb{Z}_q$ to integers $[-(q-1)/2, (q-1)/2]$ in the natural way;
- define com' to be exactly like com, but where the committed values are restricted to $[-R, R]$, where $R \ll (q-1)/2$ is some bound.

Intuitively, the bound R is here to ensure that we will have enough "room" to guarantee that if a relation between elements of $[-R, R]$ holds modulo q, then it must also hold over the integers. Looking ahead, for building a range proof, we will want to prove relations of the form $x = \sum_i x_i^2$, and we will choose R such that no overflow occurs when computing $\sum_i x_i^2 \bmod q$ with $x_i \in [-R, R]$.

The next step is to equip this commitment com' with a zero-knowledge proof system allowing to prove relations between committed values *over the integers*. However, this turns out to be particularly challenging. To see this, consider the standard Σ-protocol between a prover P and a verifier V for proving knowledge of an opening (m, r) to a commitment $c = \mathsf{com}(m; r)$:

- P: pick $(m', r') \xleftarrow{\$} \mathbb{Z}_q^2$ and send $c' = \mathsf{com}(m'; r')$.
- V: send a challenge $e \xleftarrow{\$} \mathbb{Z}_q$.
- P: send $d_m = em + m'$ and $d_r = er + r'$.

– V: accept if $\mathsf{com}(m;r)^e \cdot \mathsf{com}(m';r') = \mathsf{com}(d_m;d_r)$.

Using a standard rewinding argument, we can extract a valid opening $(m;r) \in \mathbb{Z}_q^2$ of c from any (potentially malicious) prover P^* which produces accepting proofs with non-negligible probability ε: run P^* to get c', fork it, and run it on two different random challenges e, e', receiving (d_m, d_r) and (d'_m, d'_r). By a standard probability lemma (see the splitting lemma from [PS96,PS00]), (c', e, d_m, d_r) and (c', e', d'_m, d'_r) will both be accepting transcript with non-negligible probability $\Omega(\varepsilon^2)$. From the two accepting equations, one gets

$$c = \mathsf{com}((d_m - d'_m) \cdot (e - e')^{-1}, (d_r - d'_r) \cdot (e - e')^{-1}). \tag{1}$$

To adapt the protocol to com', we would need to modify the Σ-protocol such that it additionally guarantees that the extracted value m belongs to $[-R, R]$. This actually seems feasible at first sight if we agree to settle for a relaxed correctness and zero-knowledge guarantee: we only enforce correctness and (honest-verifier) zero-knowledge whenever m belongs to $[-R', R']$, for a bound R' such that $2^{\lambda+\kappa}R' \leq R$, where κ is a statistical security parameter for zero-knowledge, and λ is a statistical security parameter for soundness (we keep both separate for generality). Then, we can modify the protocol as follows:

– P: pick $(m', r') \xleftarrow{\$} [-2^{\lambda+\kappa}R', 2^{\lambda+\kappa}R'] \times \mathbb{Z}_q$ and send $c' = \mathsf{com}(m';r')$.
– V: send a challenge $e \xleftarrow{\$} [1, 2^\lambda]$.
– P: send $d_m = em + m'$ and $d_r = er + r'$.
– V: accept if $\mathsf{com}(m;r)^e \cdot \mathsf{com}(m';r') = \mathsf{com}(d_m;d_r)$ and $d_m \in [-R, R]$.

Intuitively, relaxed correctness and relaxed statistical zero-knowledge follow from the fact that for $m \in [-R', R']$ and $e \in [1, 2^\lambda]$, $d_m = em + m'$ for $m' \xleftarrow{\$} [-2^{\lambda+\kappa}R', 2^{\lambda+\kappa}R']$ will be $2^{-\kappa}$-close to uniform (in statistical distance) over $[-R, R]$. It remains to analyze whether we can extract from an accepting prover a valid witness for com'. However, even though we restricted e and d_m to be small, recall that the extracted value (Equation 1) is of the form $m = (d_m - d'_m) \cdot (e - e')^{-1} \bmod q$. That is, m is not an element of $[-R, R]$ in general; rather, it is the product of an element in $[-R, R]$ and *the inverse modulo q of an element in* $[1, 2^\lambda]$. Therefore, this approach fails at binding the prover to a value $m \in [-R, R]$.

We note that the failure of this approach – the impossibility of extracting values guaranteed to be short in general – is a well-known problem in the context of lattice-based cryptography. Indeed, standard Σ-protocol for proving knowledge of a short solution to a system of equation – i.e., a witness for the SIS problem – suffer from exactly the same limitation (see e.g. the discussions in [BCK+14]). The standard solution is to restrict the challenge set to $\{-1, 0, 1\}$ (to guarantee that the inverse of the difference between distinct challenges remains small), and to amplify soudness via parallel repetitions. However, in our context, this would lead to a very inefficient proof system. Unfortunately, finding a different proof system with much better efficiency seems to be a hard problem.

2.2 Encoding Integers as mod-q Rationals

Instead, we follow a different approach by turning the problem around: rather than searching an efficient and sound proof system for the commitment com' above, we seek to find a different construction of bounded integer commitment $\overline{\mathsf{com}}$ such that the above efficient proof system – which is not sound because it only allows extracting fractions of small values modulo p – becomes a sound proof system for $\overline{\mathsf{com}}$ (allowing to extract bounded integers committed with $\overline{\mathsf{com}}$). Abstracting out, we saw above that we can extract from a cheating prover a triple $(y, d, \rho) \in [-R, R] \times [1, 2^\lambda] \times \mathbb{Z}_q$ such that $c = \mathsf{com}(y \cdot d^{-1} \bmod q; \rho)$. Our goal will be to find an appropriate choice of *encoding* Encode satisfying the following properties:

- $\overline{\mathsf{com}}(x; \rho) = \mathsf{com}(\mathsf{Encode}(x); \rho)$, such that a commitment to a value x' with com can be seen as a commitment to some different value $x = \mathsf{Decode}(x')$ with $\overline{\mathsf{com}}$.
- Extracting a tuple $(y, d, \rho) \in [-R, R] \times [1, 2^\lambda] \times \mathbb{Z}_q$ should correspond to extracting a valid opening of $\overline{\mathsf{com}}$ to some *bounded integer* x in an appropriate bounded range.

Looking ahead, we will need a few additional properties to hold for Encode if we want to build an efficient range proofs for $\overline{\mathsf{com}}$.

- First, we want Encode to satisfy some appropriate homomorphic properties. Informally: $\mathsf{Encode}(-x) = -\mathsf{Encode}(x)$, $\mathsf{Encode}(x + a) = \mathsf{Encode}(x) + a$, and $\mathsf{Encode}(a \cdot x) = a \cdot \mathsf{Encode}(x)$, for a sufficiently small integer a.
- Second, we want to be able to transfer a square decomposition from encodings modulo q to encoded integers: informally, proving a relation of the form $x' = \sum_i (x'_i)^2 \bmod q$ where $x' = \mathsf{Encode}(x)$ and $x'_i = \mathsf{Encode}(x_i)$ should guarantee that $x = \sum x_i^2$ *over the integers*.

Our choice of encoding. It turns out that there is a choice of (randomized) encoding that satisfies all of the above constraints simultaneously. In hindsight, this encoding is quite simple and natural: we view any pair $(y, d) \in [-R, R] \times [1, 2^\lambda]$ as an encoding $(y, d) = \mathsf{Encode}(x)$ of the integer

$$x = \left\lfloor \frac{y}{d} \right\rceil \in [-R, R],$$

where the fraction denotes standard division, and $\lfloor \cdot \rceil$ denotes *rounding to the nearest integer*. Given this choice of encoding, $\overline{\mathsf{com}}$ is defined as follows:

- $\overline{\mathsf{com}}(x)$: pick $\rho \xleftarrow{\$} \mathbb{Z}_q$ and output commitment $c = \mathsf{com}(x; \rho)$ and opening $(x, 1, \rho)$.
- $\overline{\mathsf{com}}.\mathsf{Verify}(c, \vec{x}, (y, d, \rho))$: check that $c = \mathsf{com}(y \cdot d^{-1}; \rho)$, $x = \lfloor y/d \rceil$, $y \in [-R, R]$, and $d \in [1, 2^\lambda]$.

Some remarks are in order. First, observe that $\overline{\mathsf{com}}(x)$ is defined exactly as $\mathsf{com}(x)$; that is, a honest commitment with $\overline{\mathsf{com}}$ is just a normal commitment with com. This is because we can view any $x \in [-R, R]$ as an encoding $(x, 1)$ of itself (since $x = \lfloor x/1 \rceil$). The only difference is that we relax the verification to accept general openings $(y, d) = \mathsf{Encode}(x)$ of x. Second, the fact that extracting a triple (y, d, ρ) in the Σ-protocol corresponds to extracting a valid opening (w.r.t. $\overline{\mathsf{com}}$) of an integer in $[-R, R]$ becomes trivially true. It remains to check two things:

1. $\overline{\mathsf{com}}$ must remain *binding* and *hiding*;
2. $\overline{\mathsf{com}}$ must satisfy some homomorphic properties that we outlined above.

$\overline{\mathsf{com}}$ is binding and hiding. That $\overline{\mathsf{com}}$ is hiding follows immediatly from the fact that com is hiding. It remains to consider binding. Suppose that an adversary finds two valid openings (y, d, ρ) and (y', d', ρ') in $[-R, R] \times [1, 2^\lambda] \times \mathbb{Z}_q$ to a commitment c; that is, $c = \mathsf{com}(y \cdot d^{-1} \bmod q; \rho) = \mathsf{com}(y' \cdot (d')^{-1} \bmod q; \rho')$. Since com itself is binding, we must have $y \cdot d^{-1} = y' \cdot (d')^{-1} \bmod q$. This last equation implies

$$yd' = y'd \bmod q \implies yd' = y'd \text{ over } \mathbb{Z} \implies \lfloor y/d' \rceil = \lfloor y'/d \rceil,$$

where the first implication holds as long as q is chosen large enough compared to R and 2^λ, i.e., $q/2 > R \cdot 2^\lambda$.

Properties of $\overline{\mathsf{com}}$. First, we check some basic homomorphic properties:

- If (y, d) encodes $x = \lfloor y/d \rceil$, then $(-y, d)$ encodes $-x$.
- If (y, d) encodes $x = \lfloor y/d \rceil$ and a is an integer such that $ya \leq R$, then $\overline{\mathsf{com}}(x)^a = \mathsf{com}(ayd^{-1})$ is a valid commitment $\overline{\mathsf{com}}(ax)$.
- If (y, d) encodes $x = \lfloor y/d \rceil$ and a is an integer such that $y + da \leq R$, then $\overline{\mathsf{com}}(x) \cdot \mathsf{com}(a) = \mathsf{com}(yd^{-1} + a) = \mathsf{com}((y + da)d^{-1})$ is a valid commitment $\overline{\mathsf{com}}(x + a)$ since $\lfloor (y + da)/d \rceil = \lfloor y/d + a \rceil = \lfloor y/d \rceil + a$.

Second, in our most optimized range proof constructions, we will reduce the task of proving that x belongs to an interval $[a, b]$ to the task of proving that $x_0 = (x - a)(b - x)$ is positive. To show the latter, we will prove that there exists three integers (x_1, x_2, x_3) such that $4x_0 + 1 = \sum_{i=1}^{3} x_i^3$; such a decomposition exists (and can be found efficiently) if and only if $x_0 \geq 0$ [Gro05]. Now, suppose we extracted encodings $(y, d), ((y_i, d)_{i \leq 3})$ to $4x_0 + 1$ and (x_1, x_2, x_3) respectively, with the following guarantee: $yd^{-1} = \sum_{i=1}^{3} (y_i d^{-1})^2 \bmod q$.

Intuitively, this guarantee will be obtained by using a standard Σ-protocol to prove knowledge of a 3-square decomposition directly over commitments with com. The extracted encodings will all have a common d, because of the structure of the extraction procedure: d corresponds simply to the difference between two distinct challenges for which the prover produced an accepting transcript. The above equation can be rewritten $yd = \sum_{i=1}^{3} y_i^2 \bmod q$, which necessarily holds

over the integers (i.e., no overflow occurs) given that $3R^2 < q/2$ and $2^\lambda R < q/2$, since the values y and y_i are bounded by R and d is bounded by 2^λ. From there, dividing both sides by d^2 over the rationals, we get that y/d can we written as a sum of three squares over $\mathbb{Q}$. A simple technical lemma shows that this relation over $\mathbb{Q}$ actually suffices to guarantee $x = \lfloor y/d \rceil \in [a, b]$; we omit details in this high level overview.

Note that in related work [FSW03], a similar encoding is used to allow for homomorphic computations with bounded rationals. However in our case, bounded rationals appear as an intermediate result as extracted value $(y - y') \cdot (d - d')^{-1} \bmod q$ of the proof of knowledge. Our encoding is for small integers, hence the rounding. Also, the work [LN17] uses the fact that the extracted value is unique to construct verifiable encryption schemes. Again, the application differs.

2.3 Instantiation in the Discrete Log Setting

Equipped with a method to build bounded integer commitment schemes which satisfy some necessary properties, we turn to the problem of instantiating the construction in different settings, and building a range proof from it. In the discrete logarithm setting, we set com to be the standard Pedersen commitment scheme: $\mathsf{com}(m; r) = g^m h^r$ where (g, h) are two random generators over a group where computing discrete logarithms is hard. As for the range proof, we rely on the efficient Σ-protocol of [CPP17], adapting it to prime order group (since the scheme is described over subgroups of $\mathbb{Z}_n$ for an RSA modulus n in [CPP17]). This is a relatively standard Σ-protocol where the prover, given an opening (x, r) for a commitment $c = g^x h^r$, commits to three values (x_1, x_2, x_3) such that $4(x - a)(b - x) + 1 = \sum_i x_i^2$, and proves knowledge of openings to x, x_1, x_2, x_3 such that this relation is satisfied. We provide a detailed security analysis of the resulting protocol.

The scheme of [CPP17] already includes a standard optimization for Σ-protocols, which relies on a collision-resistant hash function to compress the first flow while preserving soundness. We introduce two important additional optimizations tailored to our setting.

First Optimization. Due to our use of a group with a large order, we can actually reduce the size of the random coins used in the Pedersen commitments, at the cost of relying on the *short-exponent* discrete logarithm assumption (DLSE). This improves the computational efficiency, but also reduces the communication when proving knowledge of an opening. Furthermore, relying on DLSE has an important consequence: while the protocol of [CPP17] has computational soundness (and statistical zero-knowledge), we get an alternative instantiation which satisfies statistical soundness (and computational zero-knowledge).

On getting range proofs with statistical soundness. This alternative instantiation is obtained by changing the commitment as follows: To commit to $m \in [-R, R]$, sample $r \xleftarrow{\$} [1, K]$ and output $g^m h^r$. Here, R is a bound on the committed

messages, and K is chosen such that the short-exponent discrete log assumption, with random exponent chosen from $[1, K]$, is believed to hold. Applying DLSE, h^r is indistinguishable from a uniformly random group element (using a standard search-to-decision reduction for DLSE in prime-order groups [KK04]). Hence, the scheme remains (computationally) hiding. Furthermore, $g^m h^r$ is perfectly binding: the probability (over the random choice of s such that $g^s = h$) that there exists (m, r, m', r') with $m' \neq m$ such that $m + sr = m' + sr'$ is negligible by the Schwartz-Zippel lemma and a union bound (when R, K are small enough).

Therefore, using our proof system with short randomness in the Pedersen commitments, with appropriate parameter adjustment to guarantee perfect binding, we obtain a range proof with statistical soundness. We note that this is an important feature: the impossibility of getting statistical soundness with Bulletproof is discussed in Section 4.6 of the Bulletproof paper [BBB+18]. In anonymous transaction schemes, statistical soundness is more important than statistical zero-knowledge, since the former is crucial for avoiding indetectable creation of coins (which would render the currency useless), while the second is only necessary to guarantee anonymity (without which the currency remains usable). Not getting statistical soundness was generally believed to be inherent to efficient range proofs, since very compact commitments require computational soundness; our method shows that it is actually possible to get competitive range proofs with statistical soundness. Note that there is also a natural instantiation of our approach using ElGamal encryption as the underlying commitment scheme. This also yields a statistically sound range proof but it is less efficient than the variant of this work.

Second Optimization. The scheme of [CPP17] relies on standard "flooding" to achieve statistical zero-knowledge: the value $e \cdot m$, where $m \in [-R, R]$ is a secret value and $e \leq 2^\lambda$ is a challenge, is masked with a random $m' \xleftarrow{s} [1, 2^{\lambda+\kappa}R]$ to ensure that $em + m'$ will be $2^{-\kappa}$-close in statistical distance to the uniform distribution over $[1, 2^{\lambda+\kappa}R]$. However, it turns out that our constraints are closely related to the constraints satisfied by several Σ-protocols in the *lattice* setting, which also deal with careful bounds on the size of secret values. Building upon this observation, we import a standard optimization of Σ-protocols in the lattice-setting, namely, the *rejection-sampling* method [Lyu12]. Using rejection sampling allows different tradeoffs between the group size, the number of repetitions of the underlying protocol, and the size of the masks used to hide secret values. We show that an appropriate choice of tradeoff allows to significantly reduce the communication complexity of our protocol.

3 Preliminaries

Notation. In this work, we generally perform calculations in $\mathbb{Z}/q\mathbb{Z}$ with representatives $\mathbb{Z}_q = [-\frac{q-1}{2}, \frac{q-1}{2}]$ for an odd modulus $q \in \mathbb{N}$, and we identify $\mathbb{Z}_q$ with $\mathbb{Z}/q\mathbb{Z}$, unless stated otherwise. Inside of flooring $\lfloor \frac{a}{b} \rfloor$ or rounding $\lfloor \frac{a}{b} \rceil = \lfloor \frac{a}{b} + \frac{1}{2} \rfloor$

operations, we generally have a, b in $\mathbb{Z}$ with division over $\mathbb{Q}$, i.e. we work with the representatives and not in $\mathbb{Z}/q\mathbb{Z}$.

For some randomized algorithm $\mathscr{A}$ with input x, we sometimes 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 the randomness was sampled accordingly. We also write $s \xleftarrow{\$} S$ for sampling s uniformly random from a finite set S or $d \xleftarrow{\$} D$ to sample d randomly according to a given probability distribution D. Further, we often 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.

Throughout, we write integers $a \in \mathbb{Z}$ in lower case letters, vectors as $\vec{a} \in \mathbb{Z}^n$ with components a_i, and matrices $\boldsymbol{A} \in \mathbb{Z}^{m \times n}$ in bold upper case letters. Computations on vectors are performed component-wise, unless stated otherwise. For example, for vectors $\vec{a} = (a_i)_{i=1..n}, \vec{b} = (b_i)_{i=1..n} \in \mathbb{Z}^n$ and scalar $x \in \mathbb{Z}$, we write $\vec{c} = \vec{a} \cdot \vec{b} = (a_i \cdot b_i)_{i=1..n}, y^{\vec{B}} = (y^{b_i})_{i=1..n}$ and $\vec{B}^y = (b_i^y)_{i=1..n}$. For some constant $c \in \mathbb{Z}$, we let by $\vec{c} = (c)_{i=1..n}$ the vector with all components equal to c.

We denote by $|x|$ the absolute value of $x \in \mathbb{R}$ and by $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$ the norms defined as $\|\vec{x}\|_1 = \sum_i |x_i|$, $\|\vec{x}\|_2 = \sqrt{\sum_i x_i^2}$, $\|\vec{x}\|_\infty = \max_i |x_i|$ for $\vec{x} \in \mathbb{R}^m$.

3.1 Commitment Schemes

A commitment scheme com with message space $\mathcal{M}_{\mathrm{com}}$, commitment space $\mathcal{C}_{\mathrm{com}}$ and opening space $\mathcal{R}_{\mathrm{com}}$ is a 3-tuple of PPT algorithms $(\mathsf{Setup}, \mathsf{Commit}, \mathsf{Verify})$ such that

- $\mathsf{com.Setup}(1^\lambda)$: outputs public parameters pp,
- $\mathsf{com.Commit}_{\mathsf{pp}}(x)$: computes a commitment $c \in \mathcal{C}_{\mathrm{com}}$ to $x \in \mathcal{M}_{\mathrm{com}}$ with its opening $d \in \mathcal{R}_{\mathrm{com}}$ and outputs the pair (c, d),
- $\mathsf{com.Verify}_{\mathsf{pp}}(c, x, d)$: verifies the commitment $c \in \mathcal{C}_{\mathrm{com}}$ to $x \in \mathcal{M}_{\mathrm{com}}$ with the opening $d \in \mathcal{R}_{\mathrm{com}}$ and outputs a bit $b \in \{0, 1\}$

Further, we require that com is (statistically) correct, and satisfies binding (i.e. it is hard to find two different openings to a commitment) and hiding (i.e. one learns nothing about x from $\mathsf{Commit}(x)$). We refer to the full version for formal definitions. Often, d consists of the randomness used in the commitment generation, but it can include other auxiliary information.

(Homomorphic) Integer Commitment Schemes. In this work, we are interested in integer commitment schemes which allow to commit to an integer $x \in \mathbb{Z}$. An integer commitment scheme has message space $\mathcal{M}_{\mathrm{com}} = \mathbb{Z}$ and allows for proving relations, such as knowledge of an opening, in a zero-knowledge manner (see section 3.2). We also establish bounded integer commitment schemes (section 4.1) where the message space is $\mathcal{M}_{\mathrm{com}} = \{x \in \mathbb{Z} \mid |x| \leq R\}$ for some upper bound R. The crucial difference between message space $\mathcal{M}_{\mathrm{com}} = \mathbb{Z}_q$ and $\mathcal{M}_{\mathrm{com}} = \{x \in \mathbb{Z} \mid |x| \leq R\}$ is: The former can have additive homomorphism (over $\mathbb{Z}_q$), but only binds to a representative of $x \in \mathbb{Z}_q$, not to an integer. The latter binds to a (bounded) integer, but has limited homomorphism (over $\mathbb{Z}$).

3.2 Zero-Knowledge Proofs

We define zero-knowledge with setup GenCRS, which generates a common reference string (CRS) $\mathsf{crs} \leftarrow \mathsf{GenCRS}(\mathsf{pp})$. In this work, we only require an unstructured CRS [13]. Let R be a NP-relation over a set X defining a (pp-dependent) NP-language $\mathscr{L} = \{x \in X \mid \exists w : \mathsf{R}(\mathsf{pp}, x, w) = 1\}$. For simplicity, we suppress the dependency on pp when it is clear. A zero-knowledge proof system for $\mathscr{L}$ is a protocol between a prover P and verifier V. We write $tr \leftarrow \langle \mathsf{P}(s), \mathsf{V}(t) \rangle$ for the transcript of an interaction where P (resp. V) has input s (resp. t) and *implicit inputs* $1^\lambda, \mathsf{pp}, \mathsf{crs}$. We write $b = \langle \mathsf{P}(s), \mathsf{V}(t) \rangle$ for the verifier's verdict b. A proof system is *public coin* if the verifier's messages are uniformly random and independent of the prover's messages, and the verifier outputs $b = \mathsf{Verify}(x, tr)$ for a PPT algorithm Verify.

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

Definition 1 (Correctness). A proof system $(\mathsf{GenCRS}, \mathsf{P}, \mathsf{V})$ for $\mathscr{L}$ has *correctness error* γ_{err}, or is γ_{err}-correct, if for every adversary $\mathscr{A}$

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

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

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

Definition 2 (HVZK). A simulator Sim for a public coin proof system $(\mathsf{GenCRS}, \mathsf{P}, \mathsf{V})$ for $\mathscr{L}$ is a PPT algorithm with input a statement x for which $(\mathsf{pp}, x, w) \in \mathsf{R}$ and implicit inputs $1^\lambda, \mathsf{pp}, \mathsf{crs}$, and output a transcript tr. Let $\mathscr{A}$ be a stateful algorithm and let

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

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

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

Definition 3 (Knowledge error). Let $(\mathsf{GenCRS}, \mathsf{P}, \mathsf{V})$ be a public coin proof system for $\mathscr{L}$. Let Ext be an *expected* polynomial time oracle algorithm (with

[13] Note that the distinction between structured and unstructured random strings is crucial in real-world applications: the former unavoidably requires either a trusted third party, or a secure distributed setup. However, the latter can be instantiated in the real-world using standard heuristic 'nothing-up-my-sleeve' methods.

oracle steps counted as one step) with implicit inputs $1^\lambda, \mathsf{pp}, \mathsf{crs}$. Let $\mathscr{A}$ be a (probabilistic) and P^* be a deterministic algorithm.

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

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

W.l.o.g. Ext let $w = \bot$ if $\mathsf{Verify}(x, tr) = 1$. The advantage of $(\mathscr{A}, \mathsf{P}^*)$ is $\mathsf{Adv}^{\mathrm{ke}}_{\mathscr{A},\mathsf{P}^*,\mathsf{V}}(\lambda) = \mathsf{Real}_{\mathscr{A}}(\lambda) - \mathsf{Ideal}_{\mathscr{A}}(\lambda)$. A proof system has knowledge error κ_{err}, if for any PPT $\mathscr{A}$, P^* we have $\mathsf{Adv}^{\mathrm{ke}}_{\mathscr{A},\mathsf{P}^*,\mathsf{V}} \leq \kappa_{\mathsf{err}} + \mathrm{negl}$.

Our definition of knowledge error is closely related to witness extended emulation [Lin03,GI08], which also requires that an extractor produces convincing transcripts. This property is trivial to achieve in our setting, but interferes with our definition of knowledge error. All of our proof systems are Σ-protocols.

Definition 4. A Σ-*protocol* Σ for relation R is an interactive three-move protocol consisting of four PPT algorithms $(\Sigma.\mathsf{Init}, \Sigma.\mathsf{Chall}, \Sigma.\mathsf{Resp}, \Sigma.\mathsf{Verify})$ between prover P holding a witness w for the statement $x \in \mathscr{L}$ and verifier V such that:

- $\Sigma.\mathsf{Init}(1^\lambda, w, x) \to (\alpha, \mathsf{st})$: On input of statement and witness (x, w) with $\mathsf{R}(x, w) = 1$, outputs a first message α and a state st.
- $\Sigma.\mathsf{Chall}(1^\lambda) \to \gamma$: Draw challenge γ uniformly from the set of challenges $[0, C]$.
- $\Sigma.\mathsf{Resp}(\mathsf{st}, \gamma) \to \omega$: On input of previous state st and challenge γ, outputs a response ω.
- $\Sigma.\mathsf{Verify}(x, \alpha, \gamma, \omega) \to b$: On input statement x and transcript α, γ, ω, accepts $(b = 1)$ or rejects $(b = 0)$.

Moreover, Σ must satisfy *correctness* and *HVZK*. As usual, the algorithms have implicit inputs $1^\lambda, \mathsf{pp}, \mathsf{crs}$.

The simulators for our Σ-protocols actually show *special HVZK*, that is, they work given any (adversarial) challenge γ. Letting Sim pick $\gamma \xleftarrow{\$} [0, C]$ yields standard HVZK. To prove knowledge extraction, we rely on k-special soundness.

Definition 5 (k-special soundness). A k-special soundness extractor Ext is a PPT algorithm which takes as input a set of k accepting transcripts $\Gamma = \{(\alpha, \gamma_i, \omega_i) \mid \Sigma.\mathsf{Verify}(x, \alpha, \gamma_i, \omega_i) = 1\}_{i=1..k}$ with fixed α and pair-wise distinct challenges γ_i, and outputs a valid witness $w \leftarrow \mathsf{Ext}(\Gamma)$, i.e. $\mathsf{R}(w, x) = 1$.

In security proofs, k transcripts will either yield a witness or break an assumption. Formally, we consider the language $\mathscr{L} \vee \mathscr{L}_{\mathrm{hard}}$ instead of $\mathscr{L}$. Finding k transcripts as in definition 5 is a standard (solved) problem.

Fiat–Shamir Transformation. Informally, the Fiat–Shamir transformation applied to a Σ-protocol replaces the verifier's random challenge by a hash of the initial message α, resulting in a non-interactive proof system.

Range Proofs. A range proof is essentially a zero-knowledge proof that guarantees that a committed value x resides inside a specified interval $[a, b]$. We can show so by setting $y = (b - x)(x - a)$, computing the commitment to y homomorphically from the commitment to x and the constants a, b, and showing that $y \geq 0$ in a zero-knowledge manner. The following lemma yields a strategy to show that committed integers are non-negative.

Lemma 1 (Decomposition into 3 Squares [RS86,Gro05]). *Let $y \in \mathbb{Z}$ be an integer. It holds that*

$$y \geq 0 \iff \exists \{x_i\}_{i=1..3} : 4y + 1 = \sum_{i=1..3} x_i^2$$

Further, the integers x_i can be efficiently computed. In [PS19], the runtime of finding the decomposition was improved to $\mathcal{O}(\log^2(y)/\log\log(y))$ multiplications.

3.3 Tools in the DLOG setting

Hardness Assumptions. First, we establish the hardness assumptions that our scheme in the DLOG setting is based on (see section 5). To avoid trusted setup, we assume a deterministic family $\mathbb{G} = \mathbb{G}_\lambda$ of cyclic groups with generator g_λ and known order q_λ, generated by a group generator $(\mathbb{G}_\lambda, g_\lambda, q_\lambda) = \mathsf{GenGrp}(1^\lambda)$. For notational simplicity, we leave GenGrp implicit in the rest of the work.

Definition 6 (S-Bounded DLSE and SEI Assumption). Consider a group $\mathbb{G}$ of order q with generator g. Let $S < q$. The S-bounded DLSE assumption holds if for all PPT $\mathscr{A}$ there is a negligible negl such that

$$\Pr\left[z \xleftarrow{\$} \{0..S-1\}, z' \leftarrow \mathscr{A}(g^z) \colon z = z'\right] \leq \mathrm{negl}(\lambda)$$

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

$$\left|\Pr\left[z \xleftarrow{\$} \{0..S-1\} \ : \ \mathscr{A}(g^z) = 1\right] - \Pr\left[z \xleftarrow{\$} \mathbb{Z}_{\mathsf{ord}} \ : \ \mathscr{A}(g^z) = 1\right]\right| \leq \mathrm{negl}(\lambda)$$

Throughout this work, we generally set $S = 2^{2\lambda}$. Note that DLOG assumption is equivalent to the q-bounded DLSE assumption.

Tools. Now, we introduce some lemmas and a commitment scheme that we later on utilize for constructing the bounded integer commitment and range proof.

Lemma 2 ([KK04]). *Let $\mathbb{G}$ be a group of prime order q with generator $g \in \mathbb{G}$. For $S < q/2$, the S-bounded DLSE and SEI assumptions are equivalent.*

We consider a Pedersen commitment scheme [Ped92] with smaller openings in exchange for a computational (instead of statistical) hiding property.

Definition 7 (Pedersen Commitments with Short Openings.). Let $\mathbb{G}$ be a group of prime order q and consists of a 3-tuple of PPT algorithms $(\mathsf{Ped.Setup}, \mathsf{Ped.Commit}, \mathsf{Ped.Verify})$ such that

- $\mathsf{Ped.Setup}(1^\lambda)$: samples $g, h \xleftarrow{\$} \mathbb{G}$ and outputs public parameters $\mathsf{pp} = (g, h)$,
- $\mathsf{Ped.Commit}_{\mathsf{pp}}(x)$: samples $d \xleftarrow{\$} [0, 2^{2\lambda}]$ for $x \in \mathbb{Z}_q$, sets $c = g^x h^d$ and outputs the pair (c, d),
- $\mathsf{Ped.Verify}_{\mathsf{pp}}(c, x, d)$: outputs 1 iff $c = g^x h^d$.

Using $d \xleftarrow{\$} [0, 2^{2\lambda}]$ instead of $d \xleftarrow{\$} [0, q-1]$ (as in [Ped92]) still achieves computational hiding: Under SEI (or equivalently DLSE), we can replace the short random exponent d in h^d with a full random $d \xleftarrow{\$} [0..q-1]$ in a hybrid game. Now $g^x h^d$ is uniformly distributed, independent of x.

3.4 Tools for zero-knowledge

As a technical tool for achieving zero knowledge, our protocols use additive masking of the witness. We recall the tools for masking here.

Lemma 3 (Masking with the Security Parameter). *For any $C, B, L \in \mathbb{N}$ and fixed $x \in [-B, B], \gamma \in [-C, C]$, the distributions $U = \mathcal{U}[0, BCL]$ and $V = \{m + \gamma \cdot x \mid m \xleftarrow{\$} [0, BCL]\}$ have statistical distance at most $1/L$.*

Rejection sampling and Gaussian noise allow to use smaller masks.

Definition 8 (Discrete Gaussian Distributions, [YAZ+19]). The continuous Gaussian distribution over $\mathbb{R}^m$ centered around $\vec{v} \in \mathbb{R}^m$ with standard deviation σ is defined by the density function $\rho^m_{\vec{v},\sigma}(\vec{x}) = (\frac{1}{\sqrt{2\pi\sigma^2}})^m e^{\frac{-\|\mathbf{x}-\mathbf{v}\|_2^2}{2\sigma^2}}$. The discrete Gaussian distribution over $\mathbb{Z}^m$ centered around $\vec{v} \in \mathbb{Z}^m$ with standard deviation σ is defined as $D^m_{\vec{v},\sigma}(\vec{x}) = \rho^m_{\vec{v},\sigma}(\vec{x})/\rho^m_\sigma(\mathbb{Z}^m)$, where $\rho^m_\sigma(\mathbb{Z}^m) = \sum_{x \in \mathbb{Z}^m} \rho^m_\sigma(x)$. We write $D^m_\sigma(\vec{x}) = D^m_{\vec{0},\sigma}(\vec{x})$ for short.

Lemma 4 (Relationship between norms). *For $v \in \mathbb{R}^m$, the inequalities of norms, $\|v\|_\infty \le \|v\|_1 \le \sqrt{N}\|v\|_2 \le N\|v\|_\infty$, are well known.*

Lemma 5 (Lemma 4.4, [Lyu12]).

- *For any $k > 0$ it holds that $\Pr[|z| > k\sigma \mid z \xleftarrow{\$} D_\sigma] \le 2e^{\frac{-k^2}{2}}$.*
- *For any $k > 1$ it holds that $\Pr[\|\vec{z}\|_2 > k\sigma\sqrt{m} \mid \vec{z} \xleftarrow{\$} D^m_\sigma] < k^m e^{\frac{m}{2}(1-k^2)}$.*

Lemma 6 (Theorem 4.6, [Lyu12]). *Let V be a subset of $\mathbb{Z}^m$ in which all elements have $\|\cdot\|_2$ norms less than T, $\sigma \in \mathbb{R}$ such that $\sigma = \omega(T\sqrt{\log m})$ and $h : V \mapsto \mathbb{R}$ a probability distribution. Define algorithms $\mathscr{T}$ (resp. $\mathscr{S}$) as follows:*

1. *$\vec{v} \xleftarrow{\$} h$*
2. *$\vec{t} \xleftarrow{\$} D^m_{\vec{v},\sigma}$ (resp. $\vec{t} \xleftarrow{\$} D^m_\sigma$)*
3. *output $(\vec{t}, \vec{v})$ with probability $\min\left(\frac{D^m_\sigma(\vec{t})}{M \cdot D^m_{\vec{v},\sigma}(\vec{t})}, 1\right)$ (resp. with probability $1/M$)*

Then there exists a constant $M = O(1)$ such that the output distributions of $\mathscr{T}$ and $\mathscr{S}$ are within statistical distance $\frac{2^{-\omega(\log m)}}{M}$. Moreover, the probability that $\mathscr{T}$ outputs something is at least $\frac{1-2^{-\omega(\log m)}}{M}$.

Note that if $\sigma = \alpha T$ for some $\alpha > 0$, then $M = e^{13.3/\alpha + 1/(2\alpha^2)}$, the output of algorithm $\mathscr{T}$ is within statistical distance $2^{-128}/M$ of the output of $\mathscr{S}$ and the probability that $\mathscr{T}$ outputs something is at least $\frac{1-2^{-128}}{M}$ [YAZ+19,HPWZ17].

4 Integer Commitments from Rounding Fractions

In this section, we introduce bounded integer commitments and motivate the construction of range proofs based on these commitments.

4.1 Bounded Integer Commitment Scheme

We introduce a commitment scheme transformation that allows to commit to bounded integers. The core feature of this transformation is its proof-friendliness: standard Σ-protocols for proving knowledge of a square decomposition (or, more generally, any low-degree polynomial relation) with the original commitment (over a field $\mathbb{Z}_q$) can be re-interpreted (with minor adaptations) as Σ-protocols for proving knowledge of a square decomposition (resp. low-degree relation) over $\mathbb{Z}$ with respect to the transformed commitment scheme. In addition, the transformation preserves some homomorphic properties of the underlying scheme, which turns out to be crucial in the application to range proofs.

Definition 9 (The Transformation). Let com be a commitment scheme with message space $\mathsf{com}.\mathcal{M}_{\mathrm{com}} = \mathbb{Z}_q^n$ and random space $\mathsf{com}.\mathcal{R}_{\mathrm{com}}$. We define the commitment scheme $\overline{\mathsf{com}}$ over parameters $U, C \in \mathbb{N}$ such that $U < \frac{q-1}{2}$ with

- $\overline{\mathsf{com}}.\mathcal{M}_{\mathrm{com}} = \{\vec{x} \in \mathbb{Z}^n \mid \|\vec{x}\|_\infty \leq U/C\}$
- $\overline{\mathsf{com}}.\mathcal{R}_{\mathrm{com}} = \{(d, \gamma, \vec{y}) \in \mathcal{R}_{\mathrm{com}} \times \mathbb{Z} \times \mathbb{Z}^n \mid \gamma \leq C, \|\vec{y}\|_\infty \leq U/C\}$

as follows:

- $\overline{\mathsf{com}}.\mathsf{Setup}(1^\lambda)$: outputs $\mathsf{pp} \leftarrow \mathsf{com}.\mathsf{Setup}(1^\lambda)$.
- $\overline{\mathsf{com}}.\mathsf{Commit}_{\mathsf{pp}}(\vec{x})$: computes $(c, r) \leftarrow \mathsf{com}.\mathsf{Commit}_{\mathsf{pp}}(\vec{x})$ and outputs $(c, (r, 1, \vec{x}))$.
- $\overline{\mathsf{com}}.\mathsf{Verify}_{\mathsf{pp}}(c, \vec{x}, (r, \gamma, \vec{y}))$: sets $\vec{z} = \vec{y} \cdot \gamma^{-1} \mod q$ and checks $\vec{x} = \lfloor \frac{\vec{y}}{\gamma} \rceil, |\gamma| \leq C, \gamma \neq 0, \|\vec{y}\|_\infty \leq U/C, \mathsf{com}.\mathsf{Verify}_{\mathsf{pp}}(c, \vec{z}, r) = 1$ as well as $\vec{x} = \lfloor \frac{\vec{y}}{\gamma} \rceil$, where division is performed in $\mathbb{Q}^n$.

Lemma 7. *The commitment scheme $\overline{\mathsf{com}}$ is correct, binding and hiding.*

The correctness and hiding properties follow directly from the security of com. The binding property can be argued similarly.

Let $\mathscr{A}$ be a PPT adversary breaking the binding property of $\overline{\mathsf{com}}$. We design a PPT adversary $\mathscr{B}$ that breaks the binding property of com with challenger $\mathcal{C}$.

On receiving pp from the challenger $\mathcal{C}$, $\mathscr{B}$ forwards pp to $\mathscr{A}$ and receives $(c, (d_0, \gamma_0, \vec{y}_0), ((d_1, \gamma_1, \vec{y}_1), \vec{x}_0, \vec{x}_1)$. $\mathscr{B}$ sets $\vec{z}_i = \vec{y}_i \cdot \gamma_i^{-1} \mod q$ and just forwards $(c, d_0, d_1, \vec{z_0}, \vec{z_i})$ to $\mathcal{C}$. If $\mathscr{A}$ is successful, both commitments verify correctly with respect to $\overline{\mathsf{com}}$ and $\vec{x}_0 \neq \vec{x}_1$. Thus by definition of $\overline{\mathsf{com}}.\mathsf{Verify}$, the verification check for the sent openings are valid with respect to the scheme com. Note that $\|\vec{y}_i\|_\infty \leq U/C, |\gamma_i| \leq C$ for $i \in [0, 1]$. So $\|\vec{y}_i \cdot \gamma_i\|_\infty \leq U \leq \frac{q-1}{2}$. Assume for the sake of contradiction that $\vec{z}_0 = \vec{z}_1$:

$$\begin{aligned} \vec{z}_0 = \vec{z}_1 &\implies \vec{y}_0 \cdot \gamma_1 = \vec{y}_1 \cdot \gamma_0 \mod q \implies \vec{y}_0 \cdot \gamma_1 = \vec{y}_1 \cdot \gamma_0 \text{ in } \mathbb{Q} \\ &\implies \frac{\vec{y}_0}{\gamma_0} = \frac{\vec{y}_1}{\gamma_1} \text{ in } \mathbb{Q} \implies \left\lfloor \frac{\vec{y}_0}{\gamma_0} \right\rceil = \left\lfloor \frac{\vec{y}_1}{\gamma_1} \right\rceil \text{ in } \mathbb{Q} \end{aligned}$$

This contradicts $\vec{x}_0 \neq \vec{x}_1$ and thus the advantage of $\mathscr{B}$ is the same as $\mathscr{A}$.

Arguing over the Integers. Now, we motivate how to perform proofs over the integers on the example Ped. Let $\overline{\mathsf{Ped}}$ be the scheme obtained by the above transformation applied to Ped. Let $C = 2^\lambda$ determine the challenge space, $S = 2^{2\lambda}$ determine the size of the randomness and $L = 2^\lambda$ be the masking overhead. Let $2^\lambda = C < U \in \mathbb{N}$ and let q be prime with $2U < q$. Let $\mathbb{G}$ be a group of order q. For clarity, we restate the scheme:

- $\overline{\mathsf{Ped}}.\mathsf{Setup}(1^\lambda)$: outputs $\mathsf{pp} = (g, h) \xleftarrow{\$} \mathbb{G}^2$.
- $\overline{\mathsf{Ped}}.\mathsf{Commit}(\mathsf{pp}, x)$: samples $r \xleftarrow{\$} [0, S]$ and outputs $(c = g^x h^r, (r, 1, x))$.
- $\overline{\mathsf{Ped}}.\mathsf{Verify}(\mathsf{pp}, c, x, (r, \gamma, y))$: checks $g^{y \cdot \gamma^{-1}} h^r = c$ as well as $x = \lfloor \frac{y}{\gamma} \rceil$, where the division is performed in $\mathbb{Q}$, $|\gamma| \leq C, \gamma \neq 0$ and $|y| \leq U/C$.

The most essential protocol is the proof of knowledge of an opening. We now establish an unoptimized version in order to gain a basic understanding of the underlying arguments. The relation we prove is

$$\mathsf{R} = \{(c, (x, (r, \gamma, y))) \mid \overline{\mathsf{Ped}}.\mathsf{Verify}(c, x, (r, \gamma, y)) = 1\}.$$

For the correctness property, we are only interested in honest openings, so $\gamma = 1, y = x$. The proof scheme follows the conventional strategy of blinding the witnesses (x, r) with a mask. We add a size check for the masked witness to ensure the shortness of the opening. Note that the message space of $\overline{\mathsf{Ped}}$ is $\{x \in \mathbb{Z} \mid x \leq U/C\}$ but we can only perform proofs for smaller x values because the commitments need to stay binding after the masking process. In more detail, we let $B \in \mathbb{N}$ such that $2BCL \leq U/C$ and we allow for messages $|x_i| \leq B$. The following protocol proves knowledge of an opening.

- $\mathsf{Init}(c, (x \in [-B, B], r \in [0, S]))$: $m \xleftarrow{\$} [0, BCL], s \xleftarrow{\$} [0, SCL]$; outputs $d = g^m h^s$.
- $\mathsf{Chall}()$: outputs $\gamma \xleftarrow{\$} [0, C]$

– $\mathsf{Resp}(\gamma)$: sets $z = m + \gamma \cdot x, t = s + \gamma \cdot r$. Outputs (z, t)
– $\mathsf{Verify}(d, \gamma, z, t)$: checks $|z| \leq BCL$ and $g^z h^t = d \cdot c^\gamma$.

The first verification check succeeds with overwhelming probability since the probability that the random m is too close to BCL is small. The second check succeeds due to

$$g^z h^t = g^{m+\gamma \cdot x} h^{s+\gamma \cdot r} = g^m h^s \cdot (g^x h^r)^\gamma = d \cdot c^\gamma.$$

Further, lemma 3 also implies that z, t hide the witnesses x, r statistically and using $d = g^z h^t \cdot c^{-\gamma}$, a valid transcript can be computed for a given challenge γ. Thus, the scheme honest-verifier is zero-knowledge. The following soundness argument shows how to extract correct openings.

First, let $(d, \gamma, z, t), (d, \gamma', z', t')$ be two accepting transcripts with $\gamma \neq \gamma'$. Without loss of generality, we assume that $\gamma' > \gamma$. We denote $\overline{z} = z' - z, \overline{t} = t' - t$ and $\overline{\gamma} = \gamma' - \gamma$. We know that $g^{z'-z} h^{t'-t} = c^{\gamma'-\gamma}$ which directly implies $g^{\overline{z}/\overline{\gamma}} h^{\overline{t}/\overline{\gamma}} = c$. Thus, $\gamma^* = \overline{\gamma}, r^* = \overline{t}/\overline{\gamma}, y^* = \overline{z}$ and $x^* = \lfloor \frac{y^*}{\gamma^*} \rceil$ is a valid opening for c. Note that the size checks are satisfied:

$$|\gamma^*| \leq C, \qquad |y^*| \leq 2BCL \leq U/C.$$

Note that we know that x^* is short because γ^* and y^* are short, so the above protocol can already be seen as range proof that guarantees that the committed value lies in $[-2BCL, 2BCL]$. Nonetheless, this is not very satisfying yet because the slackness of $2CL = 2^{2\lambda+1}$ is very large. But the shortness of the extracted values can be used to argue in $\mathbb{Z}$ instead of $\mathbb{Z}_q$ which opens the door for more sophisticated arguments.

On Retaining Homomorphism. If the original scheme is homomorphic, the transformation retains (restricted) homomorphic properties. Firstly, if the commitments are generated honestly, the homomorphic property is retained as long as the homomorphic calculation is performed inside the bound U/C of the scheme. In case of dishonest commitments, the scheme still retains a more limited form of homomorphic properties.

If the scheme com allows for addition of constants to the committed value, the homomorphic property is retained up to overflow over the bound U/C. To illustrate, let $\vec{t} \in \mathbb{Z}_q^n$ be some constant and c a commitment to message $\vec{m} = \lfloor \vec{y}/\gamma \rceil$ with opening $(r, \gamma, \vec{y})$. Note that c commits to $\vec{y}/\gamma$ modulo q with respect to com and we can use the homomorphic operations. We have

$$(\vec{y}/\gamma) + \vec{t} = \vec{y}/\gamma + (\vec{t} \cdot \gamma)/\gamma = (\vec{y} + \vec{t} \cdot \gamma)/\gamma \mod q$$

and $\lfloor \frac{\vec{y}+\vec{t}\cdot\gamma}{\gamma} \rceil = \lfloor \vec{y}/\gamma \rceil + \vec{t} = \vec{m} + \vec{t}$. So the result of the homomorphic operation is actually exact because the additional operand does not introduce an additional error term. Note that for the opening to be correct, the norm $\left\| \vec{y} + \vec{t} \cdot \gamma \right\|_\infty$ needs to be smaller than U/C. So, enough space needs to be guaranteed to perform

homomorphic operations. The analysis for retaining multiplicative homomorphic properties for small constants is similar.

In the case of additive and multiplicative homomorphisms between dishonest commitments, there are some small error terms and thus, the properties do not translate as directly. We refer to the full version for more details.

For range proofs, the homomorphism with small constants can be used to prove the 3-square decomposition of the integer and the complications from multiplicative and the additive homomorphic error terms can be balanced out such that we can still prove the relation with the homomorphic property of the underlying schemes.

Ensuring Membership of an Interval. We use the 3 square decomposition in order to show membership of $[0, B]$. This can be extended to a range proof for interval $[a, b]$ by setting $B = b - a$. Since $\overline{\mathsf{com}}$ allows for addition of constants, the prover can show $x - a \in [0, B] \implies x \in [a, b]$. Note that the values still need to lie inside the given bounds.

We are using the 3 square decomposition to show that $x \in [0, B]$. Since the extracted x is a rounded fraction, we still need to ensure that the decomposition shows the desired range membership.

Lemma 8 (Three Square for Rounded Fractions). *Let $n, d \in \mathbb{Z}$ and $x = \lfloor \frac{n}{d} \rceil, \{x_i\}_{i=1..3} \in \mathbb{Q}$ and $B \geq 2$. Then:*

$$1 + 4\frac{n}{d}(B - \frac{n}{d}) = \sum_{i=1}^{3} x_i^2 \implies x \in [0, B].$$

Proof. A simple calculation shows that $\frac{n}{d} \in [\frac{1}{2}(B - \sqrt{B^2 + 1}), \frac{1}{2}(B + \sqrt{B^2 + 1})]$. This interval can further be bound as follows:

$$\frac{1}{2}(B + \sqrt{B^2 + 1}) = \frac{1}{2}B(1 + \sqrt{1 + \tfrac{1}{B^2}}) \leq \frac{1}{2}B(1 + 1 + \tfrac{1}{B^2}) = B + \tfrac{1}{2B}$$

A similar computation for the left bound shows that the 3-squares decomposition implies $\frac{n}{d} \in [-\frac{1}{2B}, B + \frac{1}{2B}]$. Since $B \geq 2$, we find $\frac{n}{d} \in [-\frac{1}{4}, B + \frac{1}{4}]$. Rounding leads to the desired result. (In fact, this holds even for $B = 1$.)

Further Properties. Our adapted commitment scheme and range proofs have additional useful properties.

Remark 1 (RP for com). For denominator $\gamma = 1$, $\overline{\mathsf{com}}$ coincides with com. Under this precondition, our range proofs establish $x \in [0, B]$ for also com-commitments.

Remark 2 (Positivity). Our proofs show $x \in [0, B]$. However, in many applications, proofs of positivity ($x \geq 0$) suffice. That is, B could be made into a zero-knowledge threshold (used for masking only), so that for $x > B$ no zero-knowledge guarantees hold.[14] This change is achieved by proving $1+4x = \sum_{i=1}^{3} x_i^2$. Now, soundness guarantees $x \in [0, \frac{q-1}{2}]$.

[14] In fact, masking and hence zero-knowledge degrades gracefully in the size of x.

Remark 3 (Denominators). A closer look at soundness shows, that a denominator $\gamma > 1$ leads to a rejection with probability $1 - \frac{1}{\gamma}$. Thus, the larger γ, the less likely will a (malicious) verifier succeed.

5 Range Proof in a DLOG Setting

5.1 Overview

In this section, we present the range proof in the setting of a group $\mathbb{G}$ with prime order q under the DLOG (or DLSE) assumption.[15] As basis, we use Pedersen commitments Ped, which we transform in a bounded rational commitment schemes $\overline{\mathsf{Ped}}$ as in section 4.1. Recall that the difference of Ped and $\overline{\mathsf{Ped}}$ is mostly in the interpretation of the committed values.

Our protocol reuses the structure of existing range proofs based on Pedersen commitments in the RSA setting (see [Lip03,Gro05,CPP17]). For a given commitment $c = g^x h^r$, the prover computes the square decomposition $1 + 4(b - x)x = \sum_{i=1..3} x_i^2$ and lets $x_0 = b - x$. Thus, we prove $1 + 4x_0 x = \sum_{i=1..3} x_i^2$. Note that all x_i are in the range $[0, B]$. The prover commits to $c_i = g^{x_i} h^{r_i}$ for some randomly sampled r_i for $i \in [1, 3]$, and sets $c_0 = g^b c^{-1}$. For a proof of knowledge of x_i, he computes mask commitments $d_i = g^{m_i} h^{s_i}$ (and an additional "garbage" term d), and sends them to the verifier. After receiving the challenge γ, the prover reveals $z_i = m_i + \gamma x_i$ and $t_i = s_i + \gamma r_i$ and the verifier can check whether the equation $g^{z_i} h^{t_i} = c_i^{\gamma} d_i$ holds (and an equation for the square decomposition).[16] The verifier checks the proof of knowledge and accepts only if z_i and t_i are small. As usual, if the prover can answer two different challenges $\gamma, \tilde{\gamma}$, openings can be extracted. These openings are $x_i = \frac{z_i - \tilde{z}_i}{\gamma - \tilde{\gamma}}$ with short nominator and denominator, and they satisfy the square decomposition (or DLOG is broken). This shows soundness (for $\overline{\mathsf{Ped}}$ openings), Furthermore, as we sketched in the introduction, when small exponents are used for the masking term h^y, and by adjusting the parameters, soundness can actually be proven *statistically*. In our parameter choice, however, we will optimize for efficiency and focus on computational soundness.

For zero-knowledge, the witness is blinded by the masks m_i. Since the m_i's must be small (hence are not uniform in $\mathbb{Z}_q$), we do not get perfect zero-knowledge. However, $x_i + m_i$ still statistically hides x_i. This is enough to establish (statistical) zero-knowledge by the usual "simulation by execution in reverse". The construction and proof is somewhat complicated by using *small* exponents for the masking term h^y, which consequently must be masked itself.

5.2 Parameters

Let $\mathsf{pp} = (g, h, q)$ be the public parameters of the commitment scheme Ped in group $\mathbb{G}$ with order q, let $\mathsf{H} : \{0,1\}^* \mapsto \{0,1\}^{2\lambda}$ be a collision resistant hash

[15] The optimization of the Pedersen commitment scheme with short exponents relies on the SEI, which for relevant ranges is equivalent to DLSE.

[16] In the scheme, we use a hash function to avoid having to send the mask commitments to the verifier to save space.

function, and let $[0, B]$ be the range with $B \geq 2$. Let $[0, C]$ be the challenge set. Let S be the size of small exponents in the SEI assumption, and let L be the growth factor of masked intervals due to additive noise, that is, masking $[0, B]$ results in $[0, BL]$. We define $U = 32B^2C^2L^2$ and note that it serves as an upper bound for the integers appearing in the security proof. In particular, we require $U < \frac{q-1}{2}$. The prover shows that he knows x, r committed in $c = g^x h^r = \overline{\mathsf{Ped}}.\mathsf{Commit}(x; r)$ and that $x \in [0, B]$. (Other commitments are interpreted as Ped)

5.3 Scheme

The scheme $\mathsf{RP}_{\mathsf{Log}}$ follows the structure of the line of work [Lip03,Gro05,CPP17]. We adapt the scheme to the DLOG setting and apply our encoding technique.

- $\mathsf{RP}_{\mathsf{Log}}.\mathsf{Init}(c = g^x h^r, x \in [0, B], r \in [0, S])$:
  1. compute x_i s.t. $4x(B - x) + 1 = \sum_{i=1}^{3} x_i^2$
  2. Set $r_0 = -r, x_0 = B - x$
  3. Set $c_0 = c^{-1} g^B$
  4. Set $\forall i \in [1, 3] : r_i \xleftarrow{\$} [0, S], c_i = g^{x_i} h^{r_i}$
  5. Set $\forall i \in [0, 3] : m_i \xleftarrow{\$} [0, BCL], s_i \xleftarrow{\$} [0, SCL], d_i = g^{m_i} h^{s_i}$
  6. Set $\sigma \xleftarrow{\$} [0, 4SBCL], d = h^\sigma c^{4m_0} \prod_{i=1..3} c_i^{-m_i}$
  7. Set $\Delta = H(\{d_i\}_{i=0..3}, d)$
  8. Outputs $\{c_i\}_{i=1..3}, \Delta$
- $\mathsf{RP}_{\mathsf{Log}}.\mathsf{Chall}()$: outputs $\gamma \xleftarrow{\$} [0, C]$
- $\mathsf{RP}_{\mathsf{Log}}.\mathsf{Resp}(\gamma)$:
  1. Sets $\forall i \in [0, 3] : z_i = m_i + \gamma \cdot x_i, t_i = s_i + \gamma \cdot r_i$
  2. Sets $\tau = \sigma + \gamma(\sum_{i=1..3} x_i r_i + 4x_0 r_0)$
  3. Outputs $\{z_i, t_i\}_{i=0..3}, \tau$
- $\mathsf{RP}_{\mathsf{Log}}.\mathsf{Verify}(\{c_i\}_{i=1..3}, \Delta, \gamma, \{z_i, t_i\}_{i=0..3}, \tau)$:
  1. Compute $c_0 = c^{-1} g^B$
  2. Compute $\forall i \in [0, 3] : f_i = g^{z_i} h^{t_i} c_i^{-\gamma}$
  3. Compute $f = h^\tau \cdot g^\gamma \cdot c^{4z_0} \cdot \prod_{i=1..3} c_i^{-z_i}$
  4. Check $\Delta = H(\{f_i\}_{i=0..3}, f)$
  5. Check $z_i \in [0, BC(L + 1)]$

The scheme is perfectly correct. Note that any interval $[0, T]$, where term T contains S, may be replaced by $[0, \max(q - 1, T)]$, as these masks only serve zero-knowledge and do not affect soundness, hence wraparound is not a problem. In particular, the scheme is correct, sound and HVZK if $S = q - 1$.

Theorem 1. *Suppose $L \geq 32$. The range proof $\mathsf{RP}_{\mathsf{Log}}$ for $[0, B]$ is 2-special sound with knowledge error $\frac{1}{(C+1)}$ under DLOG and CRHF assumptions.*
More precisely, for every adversary $\mathscr{A}$ with strict running time T there are adversaries $\mathscr{B}_1, \mathscr{B}_2$ with expected running time roughly $2T$ and $\mathsf{Adv}^{\mathrm{ke}}_{\mathscr{A}} \leq \frac{1}{(C+1)} + \mathsf{Adv}^{\mathrm{dlog}}_{\mathscr{B}_1} + \mathsf{Adv}^{\mathrm{crhf}}_{\mathscr{B}_2}$.

Proof. Assume we have two accepting transcripts for distinct challenges $\gamma \neq \tilde{\gamma}$ with witnesses z_i, t_i, τ and $\tilde{z_i}, \tilde{t_i}, \tilde{\tau}$ respectively. Without loss of generality, say $\gamma > \tilde{\gamma}$. We show that either we obtain a valid witness, or we break DLOG or collision resistance.

By collision resistance of H, we have $d = f = \tilde{f}$ and $\forall i \in [0,3] : d_i = f_i = \tilde{f}_i$. Denote by $\overline{a}$ the difference of $a - \tilde{a}$ for $a \in \{z_i, t_i, \tau\}$. From $f_i = \tilde{f}_i$ we find

$$g^{z_i} h^{t_i} c_i^{-\gamma} = g^{\tilde{z}_i} h^{\tilde{t}_i} c_i^{-\tilde{\gamma}} \iff g^{\overline{z_i}} h^{\overline{t_i}} = c_i^{\overline{\gamma}} \iff g^{\overline{z_i}/\overline{\gamma}} h^{\overline{t_i}/\overline{\gamma}} = c_i.$$

Thus for all $i \in [1,3]$, we have valid openings $x_i = \overline{z_i}/\overline{\gamma}$ and $r_i = \overline{t_i}/\overline{\gamma}$ for commitment c_i. For c_0, we obtain $c = g^{(\overline{\gamma} \cdot B - \overline{z_0})/\overline{\gamma}} h^{-\overline{t_0}/\overline{\gamma}}$ and therefore $x_0 = \overline{z_0}/\overline{\gamma}$ and $r_0 = \overline{t_0}/\overline{\gamma}$ is an opening to $c^{-1} g^B$. Moreover $x = B - \overline{z_0}/\overline{\gamma} = B - x_0$ is the committed value in c.

Now we turn to the square decomposition. We have

$$\begin{aligned}
f = \tilde{f} &\implies h^{\overline{\tau}} \cdot g^{\overline{\gamma}} \cdot c^{4\overline{z_0}} = \prod_{i=1..3} c_i^{\overline{z_i}} \\
&\implies h^{\overline{\tau}} \cdot g^{\overline{\gamma}} \cdot g^{4(B - \overline{z_0}/\overline{\gamma})\overline{z_0}} \cdot h^{4r \cdot \overline{z_0}} = \prod_{i=1..3} g^{x_i \cdot \overline{z_i}} h^{r_i \cdot \overline{z_i}} \\
&\implies g^{\overline{\gamma}} \cdot g^{4(B - \overline{z_0}/\overline{\gamma})\overline{z_0}} \cdot \prod_{i=1..3} g^{-x_i \cdot \overline{z_i}} = h^{-4r \cdot \overline{z_0}} \cdot h^{-\overline{\tau}} \cdot \prod_{i=1..3} h^{r_i \cdot \overline{z_i}} \\
&\implies g^{\overline{\gamma} + 4(B - \overline{z_0}/\overline{\gamma})\overline{z_0} - \sum_{i=1..3} x_i \cdot \overline{z_i}} = h^{-4r \cdot \overline{z_0} - \overline{\tau} + \sum_{i=1..3} r_i \cdot \overline{z_i}}.
\end{aligned}$$

Under the DLOG assumption (or statistically, when the exponent of h remains small enough), this forces

$$\begin{aligned}
&\overline{\gamma} + 4(B - \overline{z_0}/\overline{\gamma})\overline{z_0} - \sum_{i=1..3} x_i \cdot \overline{z_i} = 0 \mod q \\
\implies &\overline{\gamma} + 4(B - \overline{z_0}/\overline{\gamma})\overline{z_0} = \sum_{i=1..3} \overline{z_i}^2/\overline{\gamma} \mod q \\
\implies &\overline{\gamma}^2 + 4(\overline{\gamma} \cdot B - \overline{z_0})\overline{z_0} = \sum_{i=1..3} \overline{z_i}^2 \mod q
\end{aligned}$$

The final equality holds over the integers, because all values are small enough so that there is no wrap-around. More precisely: Let $K = BC(L+1)$ be the maximal (accepting) value of $|z_i|$. For the right hand side, $|\overline{z_i}| \leq |z_i| + |\tilde{z}_i| \leq 2K$ and hence $\sum_{i=1..3} \overline{z_i}^2 \leq 16K^2 \leq U < \frac{q-1}{2}$. Rewrite the left hand side as $\overline{\gamma}^2 + 4\overline{\gamma}B\overline{z_0} - \overline{z_0}^2$. Shortness follows from $|\gamma|B \leq K$ and thus $K^2 + 8K^2 + 16K^2 \leq 25K^2 \leq U < \frac{q-1}{2}$. Here we use that $25K^2 = 25(BC(L+1))^2 \leq 32(BCL)^2 = U$ since $L \geq 32$.

Since the equality holds over the integers, after dividing by $\overline{\gamma}^2$ it holds over $\mathbb{Q}$. Using $\overline{z_0} = \overline{\gamma}(B - x)$, we see that $\overline{\gamma}^2 + 4\overline{\gamma}x(\overline{\gamma}B - \overline{\gamma}x) = \sum_{i=1}^3 \overline{\gamma}^2 x_i^2$ and hence $1 + 4x(B - x) = \sum_{i=1}^3 x_i^2$ for $x = B - \frac{\overline{z_0}}{\overline{\gamma}}$. Now, lemma 8 finishes the proof. (Note that we extracted a valid opening for c.)

Theorem 2. *The proof system* $\mathsf{RP}_{\mathsf{Log}}$ *is HVZK with simulation error* $9/L$*. If* $S = q - 1$*, this holds against unbounded adversaries.*
More precisely, for every HVZK adversary $\mathscr{A}$*, there is a* SEI *adversary* $\mathscr{B}$ *with roughly the same running time as* $\mathscr{A}$*, such that* $\mathsf{Adv}^{\mathrm{hvzk}}_{\mathscr{A}} \leq 9/L + 4\mathsf{Adv}^{\mathrm{sei}}_{\mathscr{B}}$.

The proof works by simulation via "execution in reverse". That is, the simulator Sim picks random messages z_i, t_i first and lets $x_i = 0$. Then it uses the challenge to compute the messages from the first round. Due to masking, this distribution is L^{-1}-close to the real one. And due to SEI, replacing commitments to x_i by commitments to 0 is also indistinguishable. The full proof is in the full version.

5.4 Optimizations

We discuss some optimizations to either reduce the proof size or the group size.

Rejection sampling for smaller group size. In $\mathsf{RP}_{\mathsf{Log}}$, we hide the values $\gamma \cdot x_i \in [0, BC]$ by an additive uniformly random mask $z \in [0, BCL]$. So the masking has an overhead of $\log(L)$ bits. By using rejection sampling for masking, as used in the lattice setting, this overhead can be traded for a (small) correctness error. For this, we apply lemma 6 instead of lemma 3. That is, we choose the mask from a discrete Gaussian distribution with large enough standard deviation σ_x, and the prover aborts in Resp with (small) probability.

More concretely: Let the parameters for rejection sampling be standard deviation $\sigma_x = \alpha \cdot BC$ and $M = e^{13.3/\alpha + 1/(2\alpha^2)}$ for some α. Let $k = \sqrt{2\lambda}$ and let $L' = \lceil k\alpha \rceil$. Then the probability that the mask $m \leftarrow D_{\sigma_x}$ is too large (and causes verification to abort) is $O(2e^{-k^2/2}) = \mathrm{negl}(\lambda)$ by lemma 5. The protocol is adapted as follows[17]:

- In Init, sample $m_i \leftarrow D_{\sigma_x}$ for $i \in [0, 3]$ (instead of $m_i \leftarrow [0, BCL']$).
- In Resp, abort with probability $1 - \min\left(\frac{D_{\sigma_x}(z_i)}{M \cdot D_{\gamma \cdot x_i, \sigma_x}(z_i)}, 1\right)$ for $i \in [0, 3]$,
- In Verify, check $|z_i| \leq BC(L' + 1)$ for $i \in [0, 3]$ instead of $z_i \in [0, BC(L' + 1)]$.

Since $|m_i| \leq BCL'$ (and thus $|z_i| \leq BC(L' + 1)$) with overwhelming probability, the completeness is mostly affected by aborting in Resp. For the concrete value $\alpha = 256$ which implies $M \approx 1.05$, the abort probability is very small (roughly 0.05). The statistical distance between honest masking and "simulated" masked values is at most $\delta = 2^{-120}$, by lemma 6. Using this property the HVZK simulator is easily adapted and achieves simulation error $4\delta + 5L^{-1}$. (Note that s_i and σ are sampled as before.) The soundness proof uses L' but is otherwise unchanged.

To achieve non-negligible completeness, the protocol needs to be repeated, increasing computation and communication. For the Fiat–Shamir transformation, only computation increases.

[17] For more details on the technique and the proof of security, we refer to the range proof in the lattice setting of the full version. It uses rejection sampling for masking the randomness of the commitment scheme.

Lastly, note that $2U = 32(BCL')^2$ is a lower bound on the group size q. With rejection sampling, we can choose smaller L', and hence smaller q. One can use rejection sampling for the masks σ and s_i as well, but these do not affect the group size, only the communication (and the simulation error). More concretely, let $\sigma_r = \alpha \cdot SCL$ and further modify the protocol as follows:

- In Init choose $s_i \leftarrow D_{\sigma_r}$ for $i \in [0, 3]$.
- In Resp abort with probability $1 - \min\left(\frac{D_{\sigma_r}(t_i)}{M \cdot D_{\gamma \cdot r_i, \sigma_r} t_i)}, 1\right)$ for $i \in [0, 3]$.

This results in a size of $|t_i| \leq SCL'$. Also applying this to σ yields $|\tau| \leq 4SBCL'$. In the full version, we detail the concrete impact of these changes on the efficiency.

Soundness amplification for smaller group size. The soundness error of the scheme is $1/(C+1)$, and since C affects U and hence the group size, decreasing it allows smaller groups. However, to achieve negligible soundness error, multiple iterations are required, namely $\lambda/\log(C)$ iterations for a soundness error of $2^{-\lambda}$. Note that the commitments c_i only need to be sent in the first repetition and can be reused in the following ones.

Efficiency. Efficiency estimations are given in the introduction. Details on our calculations and the Python scripts used to compute the costs are given in the full version.

References

AC20. T. Attema and R. Cramer. Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. In *CRYPTO 2020, Part III*, *LNCS* 12172, pages 513–543. Springer, Heidelberg, August 2020.

BAZB20. B. Bünz, S. Agrawal, M. Zamani, and D. Boneh. Zether: Towards privacy in a smart contract world. In *International Conference on Financial Cryptography and Data Security*, pages 423–443. Springer, 2020.

BBB+18. B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018.

BBDE19. J. Blömer, J. Bobolz, D. Diemert, and F. Eidens. Updatable anonymous credentials and applications to incentive systems. In *ACM CCS 2019*, pages 1671–1685. ACM Press, November 2019.

BCDv88. E. F. Brickell, D. Chaum, I. Damgård, and J. van de Graaf. Gradual and verifiable release of a secret. In *CRYPTO'87*, *LNCS* 293, pages 156–166. Springer, Heidelberg, August 1988.

BCK+14. F. Benhamouda, J. Camenisch, S. Krenn, V. Lyubashevsky, and G. Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. In *ASIACRYPT 2014, Part I*, *LNCS* 8873, pages 551–572. Springer, Heidelberg, December 2014.

Ben. D. Benarroch. Diving into the zk-snarks setup phase. `https://medium.com/qed-it/diving-into-the-snarks-setup-phase-b7660242a0d7`.

BLLS20. J. Bootle, A. Lehmann, V. Lyubashevsky, and G. Seiler. Compact privacy protocols from post-quantum and timed classical assumptions. In *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020*, pages 226–246. Springer, Heidelberg, 2020.

Bou00. F. Boudot. Efficient proofs that a committed number lies in an interval. In *EUROCRYPT 2000*, *LNCS* 1807, pages 431–444. Springer, Heidelberg, May 2000.

CCs08. J. Camenisch, R. Chaabouni, and a. shelat. Efficient protocols for set membership and range proofs. In *ASIACRYPT 2008*, *LNCS* 5350, pages 234–252. Springer, Heidelberg, December 2008.

CDE+16. K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, et al. On scaling decentralized blockchains. In *International conference on financial cryptography and data security*, pages 106–125. Springer, 2016.

Cha90. D. Chaum. Showing credentials without identification transferring signatures between unconditionally unlinkable pseudonyms. In *AUSCRYPT'90*, *LNCS* 453, pages 246–264. Springer, Heidelberg, January 1990.

CHJ+20. H. Chung, K. Han, C. Ju, M. Kim, and J. H. Seo. Bulletproofs+: Shorter proofs for privacy-enhanced distributed ledger. Cryptology ePrint Archive, Report 2020/735, 2020. `https://eprint.iacr.org/2020/735`.

CHL05. J. Camenisch, S. Hohenberger, and A. Lysyanskaya. Compact e-cash. In *EUROCRYPT 2005*, *LNCS* 3494, pages 302–321. Springer, Heidelberg, May 2005.

CPP17. G. Couteau, T. Peters, and D. Pointcheval. Removing the strong RSA assumption from arguments over the integers. In *EUROCRYPT 2017, Part II*, *LNCS* 10211, pages 321–350. Springer, Heidelberg, April / May 2017.

CZJ+17. E. Cecchetti, F. Zhang, Y. Ji, A. Kosba, A. Juels, and E. Shi. Solidus: Confidential distributed ledger transactions via pvorm. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 701–717, 2017.

DF02. I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In *ASIACRYPT 2002*, *LNCS* 2501, pages 125–142. Springer, Heidelberg, December 2002.

FO97. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *CRYPTO'97*, *LNCS* 1294, pages 16–30. Springer, Heidelberg, August 1997.

FSW03. P.-A. Fouque, J. Stern, and J.-G. Wackers. Cryptocomputing with rationals. In *FC 2002*, *LNCS* 2357, pages 136–146. Springer, Heidelberg, March 2003.

GI08. J. Groth and Y. Ishai. Sub-linear zero-knowledge argument for correctness of a shuffle. In *EUROCRYPT 2008*, *LNCS* 4965, pages 379–396. Springer, Heidelberg, April 2008.

GMR89. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

Gro05. J. Groth. Non-interactive zero-knowledge arguments for voting. In *ACNS 05*, *LNCS* 3531, pages 467–482. Springer, Heidelberg, June 2005.

Gro11. J. Groth. Efficient zero-knowledge arguments from two-tiered homomorphic commitments. In *ASIACRYPT 2011*, *LNCS* 7073, pages 431–448. Springer, Heidelberg, December 2011.

HKR19. M. Hoffmann, M. Klooß, and A. Rupp. Efficient zero-knowledge arguments in the discrete log setting, revisited. In *ACM CCS 2019*, pages 2093–2110. ACM Press, November 2019.

HKRR20. M. Hoffmann, M. Klooß, M. Raiber, and A. Rupp. Black-box wallets: Fast anonymous two-way payments for constrained devices. *PoPETs*, 2020(1):165–194, January 2020.

HPWZ17. J. Hoffstein, J. Pipher, W. Whyte, and Z. Zhang. A signature scheme from learning with truncation. Cryptology ePrint Archive, Report 2017/995, 2017. `http://eprint.iacr.org/2017/995`.

KK04. T. Koshiba and K. Kurosawa. Short exponent Diffie-Hellman problems. In *PKC 2004*, *LNCS* 2947, pages 173–186. Springer, Heidelberg, March 2004.

Lin03. Y. Lindell. Parallel coin-tossing and constant-round secure two-party computation. *Journal of Cryptology*, 16(3):143–184, June 2003.

Lip03. H. Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In *ASIACRYPT 2003*, *LNCS* 2894, pages 398–415. Springer, Heidelberg, November / December 2003.

LM19. R. W. F. Lai and G. Malavolta. Subvector commitments with application to succinct arguments. In *CRYPTO 2019, Part I*, *LNCS* 11692, pages 530–560. Springer, Heidelberg, August 2019.

LN17. V. Lyubashevsky and G. Neven. One-shot verifiable encryption from lattices. In *EUROCRYPT 2017, Part I*, *LNCS* 10210, pages 293–323. Springer, Heidelberg, April / May 2017.

Lyu12. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT 2012*, *LNCS* 7237, pages 738–755. Springer, Heidelberg, April 2012.

MIO18. A. MIOLA. Addressing privacy and fungibility issues in bitcoin: confidential transactions. 2018.

Ped92. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO'91*, *LNCS* 576, pages 129–140. Springer, Heidelberg, August 1992.

PS96. D. Pointcheval and J. Stern. Security proofs for signature schemes. In *EUROCRYPT'96*, *LNCS* 1070, pages 387–398. Springer, Heidelberg, May 1996.

PS00. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.

PS19. P. Pollack and P. Schorn. Dirichlet's proof of the three-square theorem: An algorithmic perspective. *Math. Comput.*, 88(316):1007–1019, 2019.

RS86. M. O. Rabin and J. O. Shallit. Randomized algorithms in number theory. pages S239–S256, 1986.

Sle. G. Slepak. How to compromise zcash and take over the world. `https://blog.okturtles.org/2016/09/how-to-compromise-zcash-and-take-over-the-world/`.

YAZ+19. R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO 2019, Part I*, *LNCS* 11692, pages 147–175. Springer, Heidelberg, August 2019.