Vector Commitments over Rings and Compressed Σ-Protocols

Thomas Attema[1,2,3] and Ignacio Cascudo[4] and Ronald Cramer[1,2] and Ivan Damgård[5] and Daniel Escudero[6]

Abstract. Compressed Σ-Protocol Theory (CRYPTO 2020) presents an "alternative" to Bulletproofs that achieves the same communication complexity while adhering more elegantly to existing Σ-protocol theory, which enables their techniques to be directly applicable to other widely used settings in the context of "plug & play" algorithmics. Unfortunately, their techniques are restricted to arithmetic circuits over *prime* fields, which rules out the possibility of using more machine-friendly moduli such as powers of 2, which have proven to improve efficiency in applications. In this work we show that such techniques can be generalized to the case of arithmetic circuits modulo *any* number. This enables the use of powers of 2, which can prove to be beneficial for efficiency, but it also facilitates the use of other moduli that might prove useful in different applications.

In order to achieve this, we first present an instantiation of the main building block of the theory of compressed Σ-protocols, namely compact vector commitments. Our construction, which may be of independent interest, is homomorphic modulo *any* positive integer m, a result that was not known in the literature before. Second, we generalize the Compressed Σ-Protocol Theory from finite fields to $\mathbb{Z}_m$. The main challenge here is ensuring that there are large enough challenge sets as to fulfill the necessary soundness requirements, which is achieved by considering certain ring extensions. Our techniques have application as publicly verifiable zero knowledge proofs of correct computation on homomorphically encrypted data, where for a more flexible parameter instantiation it is useful that the ciphertext space is allowed to be a modular or Galois ring rather than a field: concretely, our protocols can be plugged as a commit-and-proof argument into a recent result on efficient verifiable computation schemes on encrypted data with context-hiding (PKC 21) which exploited this advantage.

1 Introduction

Zero knowledge proofs, introduced in [35], constitute an important tool used all across cryptography to build several other powerful constructions, and they also find applications outside cryptography thanks to their considerable flexibility and high potential. In a nutshell, a zero knowledge proof enables a *prover* to convince a *verifier* that a given statement belongs to certain language, without revealing anything else beyond this

fact. In addition, in a zero knowledge *proof of knowledge* the verifier gets convinced that the prover actually knows certain information, without leaking the information itself.

Given the generic nature of zero knowledge proofs, several applications and uses of these tools have been found, both inside and outside cryptography. For example, zero knowledge proofs are used thoroughly in several cryptographic constructions such as secure multiparty computation and other distributed protocols to prove, without leaking sensitive information, that certain messages are "well formed" (e.g. [33,11]). In many cases this turns out to be essential to be able to support "active adversaries", which model real-world attackers who can deviate from the specification of the cryptographic construction at hand. Furthermore, thanks to a rich and fruitful series of works [32,36,15,17,20,2,42,10,44], several zero knowledge protocols with a wide range of desirable properties and tradeoffs exist today. Quite interestingly, many of these techniques are being applied to real-world tasks, with an example being the case of ZCash [9], a digital currency that uses zero knowledge proofs to protect privacy of transactions.

Typically, zero knowledge techniques operate by somehow translating general statements to *arithmetic* statements, ultimately dealing with additions and multiplications over some algebraic structure. Traditionally, this arithmetic happens over what is known as a *finite field*, which is a set with addition and multiplication operations where every non-zero element has a multiplicative inverse. As an example, the set $\mathbb{Z}_p$, which stands for the integers modulo p, is a field when p is a prime. The tendency to use this type of structures is also present in other areas such as secure multiparty computation [24,1,26,19], and in essence, this is due to the fact that these structures possess very nice properties that make them "easy" to work with.

Finite fields, on top of being simple and well-structured algebraic constructions, can be used already in a wide range of applications. For instance, just the set $\{0, 1\}$ with the `XOR` and `AND` operations is a finite field ($\mathbb{Z}_2$, integers modulo 2), so any binary circuit as traditionally known from electrical engineering can be expressed in terms of arithmetic over the field $\mathbb{Z}_2$. Additionally, by choosing p to be large enough so that wraparound modulo p does not occur, the set $\mathbb{Z}_p$ can be used to emulate integer arithmetic, which facilitates numerical applications. However, from a mere

use-case standpoint, the choice of arithmetic modulo a prime number may seem a bit arbitrary; after all, what is so special about prime numbers? [1]

Depending on the context, other moduli may be considered equally or perhaps even more important. A natural example is the case of arithmetic modulo powers of two like 2^{64} or 2^{128}, since this corresponds to the type of basic arithmetic performed by arithmetic logic units and is expected to lead to improvements in efficiency, as is the case for secure multiparty computation [24,26]. Some other examples may include moduli structured in specific ways, such as RSA integers $N = p \cdot q$ for large prime numbers p and q, and variants of this, which could benefit applications making use of these constructions. Finally, we observe that, in mathematics, it is customary and quite enlightening to gradually reduce/abstract the required properties of a given construction to see, in essence, what are the features or patterns that enable certain propositions or constructions to hold. It is in this direction that it becomes natural to wonder if, nice and well-behaved algebraic structures such as finite fields, are really "necessary" within the context of zero knowledge proofs, or if they are simply more "convenient" to deal with.

Compressed Σ-protocols. The umbrella term "zero knowledge proofs" comprises a lot of different techniques that, although aimed at solving essentially the same problem, may drastically differ in several metrics such as efficiency, security level, underlying computational assumption, and many other. Of particular importance among these techniques, however, lies the concept of Σ-protocols [22] These tools constitute *honest verifier zero knowledge proofs of knowledge*, meaning that they enable a verifier to be convinced that a prover knows certain secret data, and this data is not leaked assuming that the verifier behaves honestly. Σ-protocols have proven to be an essential tool for building more complex protocols, like actual malicious verifier zero knowledge proofs, but also more elaborate systems such as proofs of disjunctions and proofs of some-out-of-many statements [25], identification schemes [41], among many others. They have also been used in other contexts such as maliciously secure multiparty computation with a dishonest majority (e.g. [11]).

In [3], the authors presented a series of techniques for *compressing* Σ-protocols, in a way that adheres to the existing theory of Σ-protocols and therefore inherits all the results and applications from the field. Other

[1] Of course, within mathematics, prime numbers hold a special throne, but from an application point of view modular arithmetic is essentially the same regardless of the chosen modulus.

works such as [17] achieved similar results in terms of communication efficiency, but were presented as a replacement for standard Σ-protocol theory and, as a result, do not serve as a building block for constructions making use of Σ-protocols, or at least not without any (typically non-trivial) adaptation.

The results in [3] shed an important light on the expressibility and efficiency of the Σ-protocol framework. However, as is the case with most of the literature on interactive proofs and zero knowledge proofs, their techniques are restricted to finite fields, which is made evident from the fact that they use several tools restricted to finite fields such as polynomial interpolation or Pedersen commitments, among others. Given the importance of this general theory, a worthy goal is then to extend the results in [3] to the setting in which the algebraic structure under consideration is not necessarily a finite field $\mathbb{Z}_p$. This would enable the use of these tools in a much wider range of applications and scenarios, and it could also potentially boost its efficiency by considering rings of the form $\mathbb{Z}_{2^k}$. In addition, as discussed earlier, such study would make more clear what is the inherent reach and limitation of the theory on compressed Σ-protocols, in terms of the underlying algebraic structure.

1.1 Our Contribution

In this work we explore an extension of the compressed Σ-protocol framework from [3], from the case in which the algebraic structure is a field of the form $\mathbb{Z}_p$, to the more general setting of $\mathbb{Z}_m$, for an *arbitrary* positive integer m. In a nutshell, our results show that compressed Σ-protocols for partial openings over $\mathbb{Z}_m$, where a prover shows that it knows how to open a commitment to a vector that maps to a given value under certain $\mathbb{Z}_m$-linear map, are possible. This is achieved in a direct and efficient manner, without the need to "emulate" arithmetic using existing field-based techniques.

Finally, our techniques inherit all the "plug & play" applications of [3], and in particular, they can be used in a wide range of settings in which Σ-protocols prove useful, without the restriction of having a prime modulus. As an example of this, we show in Section 5 an application to the domain of efficient verifiable computation schemes on encrypted data, where the recent work of [14] offered a flexible framework that can deal efficiently with the general case in which the ciphertext space of the homomorphic encryption scheme is a polynomial ring with coefficients in a ring $\mathbb{Z}_m$. Their variant with context-hiding, where the verifier is not allowed to learn information about the inputs of the computation (including the

possible inputs from the server doing the computation), requires commit-and-prove arguments for certain statements defined over modular and Galois rings. They leave open the existence of succinct arguments that work directly over rings. Given that the majority of these statements are linear maps over $\mathbb{Z}_m$ (the remaining being range proofs, which are also easily dealt with using compressed Σ-protocols), our results seem much better suited for this application than using an argument that works over a field and having to emulate the arithmetic over $\mathbb{Z}_m$, which would introduce more non-linear conditions.

A detailed overview of our techniques is presented in Section A in the Supplementary Material. At a high level, our results are obtained as a combination of two main contributions, discussed below.

Compact Vector Commitments over $\mathbb{Z}_m$. One of the core ingredients in the context of zero-knowledge proofs, and in particular [3], are *commitment schemes*. These must be homomorphic over the given algebraic domain, which is $\mathbb{Z}_m$ for an arbitrary integer m in our setting. In [3] different instantiations of this construction are considered, namely Pedersen commitments and also RSA-based commitments. However, these constructions are restricted to m being a prime, and, besides a few exceptions that will be discussed in Section 1.2 below, no construction of a compact vector commitment scheme with homomorphism over $\mathbb{Z}_m$ for an arbitrary m is known. To tackle this issue we present in Section 3, as a contribution of potential independent interest, an efficient construction of said commitment schemes. This is achieved by first abstracting and generalizing a template present in several previous schemes like Pedersen's to obtain a compact vector commitment scheme from a single-value construction, and then focusing on instantiating the latter type of commitments. To this end, depending on the parity of m, we either rely on the hardness of finding roots over RSA groups, or factoring.

Compressing Mechanism over $\mathbb{Z}_m$. In order to compress a basic three-move Σ-protocol, the work of [3] resorts to using an efficient proof of knowledge to handle the last message in such a protocol, which constitutes the prover's response to the verifier's challenge. In [3], the proof of knowledge used is an adaptation of Bulletproof's folding technique [15,17]. This is not restricted to finite fields per se, but it does require large enough *exceptional sets*, also known as *challenge sets*, for it to obtain reasonably small soundness error. If m is prime, and in general, if m does not have small prime factors, then such sets over $\mathbb{Z}_m$ exist, but if m is divisible by a small prime then this does not hold. To address this issue, we resort to

considering ring extensions of the form $\mathbb{Z}_m[X]/(f(X))$ for a polynomial $f(X)$, which increases the sizes of the required exceptional sets. We show in Section 4 that our commitment construction is compatible with this type of arithmetic, and that this leads to a natural adaptation of the results from [3] from the field setting to $\mathbb{Z}_m$, for an arbitrary m.

1.2 Related Work

Compressed Σ-protocol theory [3], which we take as a starting point, presents a Σ-protocol for proving knowledge that a vector underlying a given commitment satisfies certain linear relation. The linear communication complexity of this Σ-protocol is then compressed down to logarithmic by adapting the techniques from [15,17]. As we have already mentioned, the techniques in the references cited above are mostly suitable when the computation domain is a finite field $\mathbb{Z}_q$.

An instantiation of compressed Σ-protocol theory in the context of lattices is presented in [4]. Lattice-based (compressed) Σ-protocols allow provers to prove knowledge of a *short* homomorphism preimage, i.e., a preimage of bounded norm. However, these protocols have the additional complication that the norm bound β of the secret witness, known by an honest prover, differs from the norm bound $\tau \cdot \beta$ that the prover ends up proving. The factor τ is referred to as the *soundness slack*. In most practical scenarios, this relaxed functionality is sufficient. However, due to the soundness slack, lattice-based compressed Σ-protocols have polylogarithmic, instead of logarithmic, communication complexity. These complications would be attenuated by using ring extensions as we do here, so their techniques do not directly fit our purpose.

Further, [18] presents an adaptation of Bulletproofs defined over the integers $\mathbb{Z}$. Their techniques allow a prover to prove knowledge of a vector of *bounded* integers satisfying arbitrary constraints captured by a circuit over $\mathbb{Z}$. However, Block et al. [13] recently found a gap in the analysis of [18]. A non-trivial adaptation, increasing the communication complexity from logarithmic to polylogarithmic, was required to overcome this issue [13].

By appropriately encoding vectors $\mathbf{x} \in \mathbb{Z}_m^n$ as (bounded) integers, we thus obtain a zero-knowledge proof system for relations defined over the ring $\mathbb{Z}_m$ for an arbitrary $m \in \mathbb{N}$. However, this indirect approach results in polylogarithmic communication complexity, while our construction works directly over $\mathbb{Z}_m$ and achieves $\mathcal{O}(\log n \log\log n)$ communication complexity. Moreover, it cannot harness the efficiency improvements foreseen when the arithmetic takes place in rings $\mathbb{Z}_m$, with $m = 2^{64}$ or

$m = 2^{128}$, corresponding to machine computations. These efficiency improvements have already been demonstrated in multiparty computation applications [24,26],

Zero knowledge for more general rings than $\mathbb{Z}_q$ for a prime q has not been studied in great detail, to the best of our knowledge. In this direction, the only works we are aware of are Rinocchio [30], which presents a succinct non-interactive arguments of knowledge (SNARK) protocol for statements represented as circuits over general commutative rings having large enough exceptional sets, and the "Appenzeller to Brie" zero-knowledge protocol from [7]. None of these two works are based on Σ-protocols.

Finally, in terms of homomorphic and compact vector commitments, to the best of our knowledge, no previous work has tackled the case in which the underlying algebraic structure is $\mathbb{Z}_m$, for an arbitrary m. Most existing constructions only work for m a prime, as is the case for example with Pedersen commitments [40] and also constructions based on homomorphic encryption such as ElGamal [27]. Furthermore, some schemes such as Paillier [39] or Okamoto-Uchiyama [38] operate over non-prime modulus, but these are still very structured (e.g., $N = PQ$ or $N = P^2Q$). Homomorphic commitments over $\mathbb{Z}_{2^k}$ exist, such as the Joye-Libert construction [37], but it is not clear how to generalize this approach to powers of odd primes. Even most lattice-based homomorphic commitments such as [8,12] require a prime modulus so that their associated algebraic structure factors nicely.

2 Preliminaries

We begin by recalling some basic notions that we will use thoughout our work. This is organized as follows. First, in Section 2.1 we present the concept of vector commitments, which are the core objects underlying our techniques and contribution. Then, in Section 2.2, we discuss the idea of an interactive protocol, and we consider concepts like completeness, soundness and zero-knowledge.

Some general notation. Before we dive into the aforementioned ideas, we introduce some general notation. Let m be a positive integer. The ring of integers modulo m is denoted by $\mathbb{Z}_m$. Vectors are denoted by bold letters, like $\mathbf{x}$ and $\mathbf{y}$, and, unless explicitly specified, their coordinates will be denoted by the same letter with normal font and a subscript representing the index, e.g. x_i and y_i. The notation $\mathbf{x} + \mathbf{y} \bmod m$ represents addition

modulo m coordinate-wise, although we remark that most of the time we will omit the "mod m" notation and assume operations are modulo m when it is clear from context. Also, given a finite set A, we denote by $a \leftarrow A$ the process of sampling a uniformly random value a from A.

2.1 Vector Commitments

At a high level, a vector commitment over $\mathbb{Z}_m^n$ enables a party to compute some data from an n-dimensional vector over $\mathbb{Z}_m$ in such a way that (1) the derived data does not reveal anything about the original vector and (2) if the party decides to "open" the vector (e.g. announce it to other parties) at a later point, then the additional computed data ensures he cannot "change his mind" by announcing a different vector. In essence, this is effectively the mathematical analogue of writing down the vector in a "paper envelope", hide it, and announce the vector in the future together with the envelope.

The formal definition is as follows.

Definition 1 (Vector commitments). *A* homomorphic vector commitment scheme *for $\mathbb{Z}_m^n$ is defined by a tuple $(\mathcal{G}, \textsc{Com}, R)$, where $\mathcal{G}$ is a probabilistic polynomial time algorithm, called the* key generation algorithm*, and $\textsc{Com}, R$ are polynomial time computable functions, satisfying the following syntax.*

- *$\mathcal{G}(m, n, \kappa)$ outputs a public key pk.*
- *$\textsc{Com}_{\mathsf{pk}}$ takes as input a vector $\mathbf{x} \in \mathbb{Z}_m^n$ and a uniformly random r sampled from a domain $\mathcal{R}$, and produces a string c. We assume that the image of $\textsc{Com}_{\mathsf{pk}}$ is a finite group, and we assume that the group operation can be computed efficiently given the public key. We use multiplication notation for this operation.*
- *R_{pk} produces as output an element of $\mathcal{R}$. It receives different possible inputs and these will be clarified below. Also, we typically omit the pk subindex, using the notation $R(\cdot)$ instead of $R_{\mathsf{pk}}(\cdot)$.*

The properties required from these algorithms are the following. Below, we let $\mathsf{pk} \leftarrow \mathcal{G}(m, n, \kappa)$.

- ***Perfect Hiding.** For any values $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}_m^n$, the distributions of $\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r)$ and $\textsc{Com}_{\mathsf{pk}}(\mathbf{x}', r')$ for uniformly random $r, r' \in \mathcal{R}$ are identical.*

– ***Computational Binding.*** *Consider the following experiment on a probabilistic polynomial time algorithm A: Sample* $\mathsf{pk} \leftarrow \mathcal{G}$ *and send this value to A, who wins the game if it outputs* $(\mathbf{x}, r, \mathbf{x}', r')$ *such that* $\mathbf{x} \neq \mathbf{x}'$ *and* $\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r) = \textsc{Com}_{\mathsf{pk}}(\mathbf{x}', r')$*. The probability (taken over the choice of* pk *and the random coins of A) that any such A wins is negligible (in* κ*).*
– ***Homomorphic property.*** *The following holds:*[2]

$$\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r) \cdot \textsc{Com}_{\mathsf{pk}}(\mathbf{x}', r') = \textsc{Com}_{\mathsf{pk}}(\mathbf{x} + \mathbf{x}', R(\mathbf{x}, \mathbf{x}', r, r'))\,,$$

$$\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r)^{-1} = \textsc{Com}_{\mathsf{pk}}(-\mathbf{x}, R(\mathbf{x}, -1, r))\,.$$

What the equations in the final homomorphic property above say is that the product of two commitments can be opened to reveal the sum modulo m of the two original values, and the inverse of a commitment can be opened to minus the original value (modulo m). Furthermore, in all cases, the committer, that is, the party who computed the commitments in a first place, can compute appropriate randomness values for the new commitments using the function R. Finally, note that, by adding a commitment to itself and using the above rules, it is implied in a natural way that c^a, for commitment $c = \textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r)$ and integer a, can be opened as $a \cdot \mathbf{x}$ (modulo m). We write the associated randomness as $R(\mathbf{x}, a, r)$, i.e., $\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r)^a = \textsc{Com}_{\mathsf{pk}}(a \cdot \mathbf{x}, R(\mathbf{x}, a, r))$.

In addition to the properties from Definition 1 above, we also require the following property:

– **Randomization property.** For any correct pk, $\mathbf{x}', \mathbf{x} \in \mathbb{Z}_m^n$, if at least one of r or r' is chosen uniformly at random in $\mathcal{R}$, then $R(\mathbf{x}, \mathbf{x}', r, r')$ is uniform in $\mathcal{R}$.

Intuitively, this property will enable us to randomize commitments by multiplying by a random commitment. In a nutshell, the idea is that, if one opens a product commitment $\textsc{Com}_{\mathsf{pk}}(\mathbf{x} + \mathbf{x}', R(\mathbf{x}, \mathbf{x}', r, r'))$, then, as long as one of r, r' is uniform, the only information this reveals on $\mathbf{x}, \mathbf{x}'$ is $\mathbf{x} + \mathbf{x}'$.

Single-Commitments. Finally, we consider the notion of a single-commitment scheme. At a high level, a single-value commitment scheme is a vector commitment scheme that only allows $n = 1$, i.e. only "vectors" of dimension 1 can be committed to. However, for our needs, we impose the following additional condition on single-value commitment schemes.

[2] Note that we allow the R-function to take both 1 (zero-openings), 3 and 4 arguments.

Definition 2. *A* single-value homomorphic commitment scheme *for* $\mathbb{Z}_m$ *is a homomorphic vector commitment scheme for values in* $\mathbb{Z}_m^n$ *that only allows* $n = 1$*, and has the additional property specified below.*

- ***Zero-commitment opening.*** *For any single-value commitment* c*,* c^m *can be opened as zero, More specifically, we have that*[3]

$$c^m = \text{COM}_{\mathsf{pk}}(0, R(c)).$$

Note that the zero-commitment opening property implies that, given a commitment c, c^m can be opened by a party who possibly did not create c in a first place. The fact that c^m is a commitment to 0 is already implied by the homomorphic property implies given that, if c is a commitment $c = \text{COM}_{\mathsf{pk}}(\mathbf{x}, r)$, it holds that $c^m = \text{COM}_{\mathsf{pk}}(m \cdot \mathbf{x}, R(\mathbf{x}, m, r)) = \text{COM}_{\mathsf{pk}}(\mathbf{0}, R(\mathbf{x}, m, r))$. However, the zero-commitment opening property ensures that this is the case, and that the corresponding randomness can be derived from c alone.

Intuitively, the reason why this property is needed is the following. The commitment schemes we consider in this work are intended to be homomorphic modulo m, meaning that their message space forms a module over $\mathbb{Z}_m$, and (linear) operations over commitments should correspond to the analogue operations over the message space. Nevertheless, we are only assuming that the set in which the commitments live is a finite group, and we do not assume anything about its order. The zero-commitment property ensures that, even though this group's exponent may not be a divisor of m (so commitments raised to the m-th power may not lead to the identity of the group), raising to the m-th power still leads to commitments that can be easily dealt with. We will make use of this property, for example, in Theorem 1 when we prove the homomorphic property of our vector commitment scheme.

2.2 Interactive Proofs

In this work we consider interactive proof that, given an NP-relation R, enable a prover to prove knowledge of a witness w with respect to a given statement x, where $(x; w) \in R$. An interactive proof is (perfectly) *complete* (or satisfies *completeness*) if for all inputs $(x; w)$, if $(x; w) \in R$, then the verifier outputs `accept` with probability 1. We also consider the notion of $(k_1, \ldots, k_\mu)$-*special soundness*, which means that, given a $(k_1, \ldots, k_\mu)$-tree of accepting transcripts and a statement x, a witness

[3] Here we, once again, abuse notation and let R take a commitment as input.

w such that $(x; w) \in R$ can be computed efficiently. Interactive proofs satisfying this notion are known to be proofs of knowledge [4]. For more details we refer to Section B in the Supplementary Material.

In this work we consider *public coin* interactive proof in which the messages sampled by the verifier are uniformly random. In regards to zero-knowledge, as typical with Σ-protocols, we restrict our attention to *special honest-verifier zero-knowledge* (SHVZK), which requires that, given a statement x and a set of uniformly random verifier messages, it is possible to produce (without knowing any witness) an accepting transcript that follows the same distribution as an actual interaction between the prover and the verifier.

3 Vector Commitments over $\mathbb{Z}_m$

In this section we present one of the main contributions of our work, namely, the construction of a compact modulo-m homomorphic vector commitment scheme. Here, compact, means that the size of a commitment is independent of the vector dimension n. Our result is achieved in two steps. First, we show in Section 3.1 a generic method to obtain a compact vector commitment scheme from *any* single-value commitment scheme, as defined in Section 2.1. Then, in Section 3.2, we present a construction of a single-value commitment scheme based on what we call commitment friendly groups. This transformation works for any value of m, and, in the same section, we present an instantiation of commitment friendly groups that, unfortunately, is restricted to odd values of m. To address this issue, we present in Section 3.3 a construction of single-value commitment schemes for the case in which m is even.

3.1 Vector Commitments from Single-value Commitments

Let m be any positive integer, and let $(\mathcal{G}', \textsc{Com}', R')$ be a single-value commitment scheme for $\mathbb{Z}_m$. The goal of this section is to derive from this scheme, for any positive integer n, a compact vector commitment scheme $(\mathcal{G}, \textsc{Com}, R)$. At a high level, our construction generalizes the approach followed in Pedersen's construction to obtain compact commitments to long vectors, by essentially taking a "random linear combination in the exponent". We present our compact vector commitment scheme in full detail below.

$\mathsf{VC}_{m,n}$: Vector Commitment Scheme for $\mathbb{Z}_m^n$

$(\mathcal{G}', \textsc{Com}', R')$ is a single-value commitment scheme for elements over $\mathbb{Z}_m$

- $\mathcal{G}$, on input n, m, κ, proceeds as follows.
  1. Run $\mathsf{pk}' = \mathcal{G}'(m, \kappa)$.
  2. For $i = 1, \ldots, n$, sample $a_i \leftarrow \mathbb{Z}_m$ and $r_i \leftarrow \mathcal{R}$. Set $g_i = \textsc{Com}'_{\mathsf{pk}'}(a_i, r_i)$
  3. Output $\mathsf{pk} = (\mathsf{pk}', g_1, \ldots, g_n)$.
- Given $\mathbf{x} = (x_1, \ldots, x_n)$ and $r \in \mathcal{R}$ as input, $\textsc{Com}_{\mathsf{pk}}$ outputs $\textsc{Com}'_{\mathsf{pk}'}(0, r) \cdot \prod_{i=1}^n g_i^{x_i}$.

As we shall see in a moment, there is a very efficient reduction that shows that the binding property holds in $\mathsf{VC}_{m,n}$, assuming that it holds on the underlying single-value commitment scheme, with only a $1/2$ factor loss (which is independent of n) in terms of success probability of the corresponding computational binding experiment. In addition, observe that the vector commitment scheme $\mathsf{VC}_{m,n}$ is compact, given that a commitment is made of a homomorphic combination of single-value commitments.

Finally, it may seem that a trusted set-up is needed in our construction, given that the public key generation algorithm $\mathcal{G}$ chooses the g_i's with known content a_i and, as we will see, the binding property depends on the a_i's being unknown. However, in some cases, including the instantiations we will present, it is possible to choose the g_i's obliviously with identical or at least indistinguishable distribution, removing this potential point of failure.

Theorem 1. *When based on a single-value homomorphic commitment scheme for $\mathbb{Z}_m$ satisfying Definition 2, $\mathsf{VC}_{m,n}$ is a homomorphic vector commitment scheme for $\mathbb{Z}_m^n$, according to Definition 1.*

Proof. To see that the perfect hiding property holds, begin by observing that, by construction of the g_i's and the homomorphic property of the single value scheme, we have

$$\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r) = \textsc{Com}'_{\mathsf{pk}'}\Big(\sum_{i=1}^n a_i x_i, s\Big) \cdot \textsc{Com}'_{\mathsf{pk}'}(0, r),$$

for some s that can be computed by applying the R-function of the single value scheme several times on inputs $\mathbf{x}$ and $r_1, \ldots, r_n$. Perfect hiding now follows immediately from the perfect hiding property of the underlying single-value scheme, together with its randomization property, which ensures that the randomness appearing in the overall commitment above is uniformly random.

For the the binding property, assume the existence of an adversary A that wins the binding experiment for $\mathsf{VC}_{m,n}$ with probability ϵ. We will show that such an adversary can be used to build an adversary B that breaks binding experiment of the original single-value scheme with probability at least $\epsilon/2$. Since ϵ is negligible, given that the underlying single-value scheme satisfies the binding property, we obtain that $\mathsf{VC}_{m,n}$ satisfies the property as well.

We define the algorithm B as follows. B gets a public key pk' as input, and then expands this to a public key $\mathsf{pk} = (\mathsf{pk}', g_1, \ldots, g_n)$ following the definition of $\mathcal{G}$. Then B runs A on input pk. Now, assume that A wins, which means that A outputs $(\mathbf{x}, r, \mathbf{x}', r')$ with $\mathbf{x} \neq \mathbf{x}'$ and $\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r) = \textsc{Com}_{\mathsf{pk}}(\mathbf{x}', r')$. As we did with the hiding property, we can write both sides of the expression above in terms of single-value commitments, as follows: the left-hand side equals $\textsc{Com}'_{\mathsf{pk}'}(\sum_{i=1}^{n} a_i x_i, s) \cdot \textsc{Com}'_{\mathsf{pk}'}(0, r)$, while the right-hand side is $\textsc{Com}'_{\mathsf{pk}'}(\sum_{i=1}^{n} a_i x'_i, s') \cdot \textsc{Com}'_{\mathsf{pk}'}(0, r')$, for values s, s' that can be efficiently computed. Using the homomorphic property of the original scheme once more, we get

$$\textsc{Com}'_{\mathsf{pk}'}\Big(\sum_{i=1}^{n} a_i x_i, R'\Big(\sum_{i=1}^{n} a_i x_i, 0, s, r\Big)\Big) = \textsc{Com}'_{\mathsf{pk}'}\Big(\sum_{i=1}^{n} a_i x'_i, R'\Big(\sum_{i=1}^{n} a_i x'_i, 0, s', r'\Big)\Big).$$

If $\sum_i a_i x_i \neq \sum_i a_i x'_i \bmod m$, this clearly means that B can break binding of the original scheme by outputting these values together with the corresponding randomness used for the commitments above. To finish the proof of our main claim, it suffices then to show that $\sum_i a_i x_i \neq \sum_i a_i x'_i \bmod m$ happens with probability at least $1/2$.

To see this, assume that $\sum_i a_i(x_i - x'_i) = 0 \bmod m$. Since we are assuming that A wins, we have $x_{i_0} - x'_{i_0} \neq 0 \bmod m$ for some i_0. From this, it must be the case that $x_{i_0} - x'_{i_0} \neq 0 \bmod p$ for at least one prime factor p in m. Additionally, notice that $\sum_{i=1}^{n} a_i(x_i - x'_i) = 0 \bmod p$, given that the corresponding congruence holds modulo m, so we can rewrite $a_{i_0} = -(x_{i_0} - x'_{i_0})^{-1} \cdot \sum_{i \neq i_0} a_i(x_i - x'_i) \bmod p$. Now, notice that by the hiding property of the single-value scheme, the g_i's included in the public key of $\mathsf{VC}_{m,n}$ follow a distribution that is independent of the a_i's, so, in particular, the $x_i - x'_i$ values produced by A are independent of these a_i's. From this, we see that the right-hand side of the previous expression is independent of the left-hand side, which is uniformly random, so the probability of this equation being satisfied is at most $1/p$, or, in other words, B wins the binding experiment with probability $1 - 1/p \geq 1 - 1/2 = 1/2$. This implies that B succeeds with an overall probability of at least $\epsilon/2$, which proves the binding property of the vector commitment scheme.

To establish the homomorphic property, consider commitments $\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r) = \textsc{Com}'_{\mathsf{pk}'}(0, r) \cdot \prod_{i=1}^{n} g_i^{x_i}$ and $\textsc{Com}_{\mathsf{pk}}(\mathbf{x}', r') = \textsc{Com}'_{\mathsf{pk}'}(0, r') \cdot \prod_{i=1}^{n} g_i^{x'_i}$. Using the homomorphic property of the single-value scheme, we can write

$$\begin{aligned}
\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r) \cdot \textsc{Com}_{\mathsf{pk}}(\mathbf{x}', r') &= \prod_{i=1}^{n} g_i^{x_i + x'_i} \cdot \textsc{Com}'_{\mathsf{pk}'}(0, r) \cdot \textsc{Com}'_{\mathsf{pk}'}(0, r') \\
&= \prod_{i=1}^{n} g_i^{x_i + x'_i} \cdot \textsc{Com}'_{\mathsf{pk}'}(0, R'(0, 0, r, r')) \\
&= \prod_{i=1}^{n} g_i^{x_i + x'_i \bmod m} g_i^{\ell_i m} \cdot \textsc{Com}'_{\mathsf{pk}'}(0, R'(0, 0, r, r')),
\end{aligned}$$

where ℓ_i is defined by $x_i + x'_i = ((x_i + x'_i) \bmod m) + \ell_i m$. Now, recall that the zero-commitment opening property from Definition 2 of the single-value commitment scheme enables, for any commitment c, to open c^m to zero. Since $g_i^{\ell_i}$ is a valid commitment (to $\ell_i \cdot a_i \bmod m$, but this is irrelevant), we have that $(g_i^{\ell_i})^m = \textsc{Com}'_{\mathsf{pk}'}(0, R'(g_i^{\ell_i}))$. Inserting this in the above is easily seen to imply that

$$\begin{aligned}
\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r) \cdot \textsc{Com}_{\mathsf{pk}}(\mathbf{x}', r') &= \prod_{i=1}^{n} g_i^{x_i + x'_i \bmod m} \textsc{Com}'_{\mathsf{pk}'}(0, R'(g_i^{\ell_i})) \cdot \textsc{Com}'_{\mathsf{pk}'}(0, R'(0, 0, r, r')) \\
&= \prod_{i=1}^{n} g_i^{x_i + x'_i \bmod m} \textsc{Com}'_{\mathsf{pk}'}(0, s) = \textsc{Com}_{\mathsf{pk}}(\mathbf{x} + \mathbf{x}', s)\,,
\end{aligned}$$

for some $s \in \mathcal{R}$ that can be computed by applying the randomness function R' of the single value scheme several times on inputs $\mathbf{x}, \mathbf{x}', r, r', g_1, \ldots, g_n$. This (implicitly) defines the randomness function R of the vector scheme. In a very similar way, one proves that $\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r)^{-1}$ can be opened as $-\mathbf{x} \bmod m$. Namely, if we insert the expression for $\textsc{Com}_{\mathsf{pk}}(\mathbf{x}, r)$, we get $-x_i$'s appearing in the exponent, but these are equal to $-x_i \bmod m$ except for a multiple of m which can "absorbed" into the randomness factor in the commitment using the zero-commitment opening property.

The randomization property follows immediately from the randomization property of the original scheme. □

3.2 Single-Value Commitments via Commitment Friendly Groups

In the previous section we show how one can obtain a compact vector commitment scheme over $\mathbb{Z}_m^n$ assuming the existence of a single-value commitment scheme over $\mathbb{Z}_m$. In this section, we proceed to present an

instantiation of one such scheme. We begin by introducing the concept of commitment friendly groups, which plays a pivotal role in our construction.

Commitment Friendly Groups. We will assume we have a probabilistic prolynomial time algorithm $\mathcal{GG}$ which, on input m and security parameter κ, outputs a finite Abelian group G.[4] For a prime p dividing m, consider the function $\phi_p \colon G \mapsto G$ given by $\phi_p(g) = g^p$, where p is a prime factor in m.

Definition 3 (Commitment friendly groups). *We say that $\mathcal{GG}$ is* commitment friendly *if for all primes $p \mid m$, the following holds:*

1. *ϕ_p is collision intractable, i.e, it is hard to find a collision: $g \neq g'$ such that $\phi_p(g) = \phi_p(g')$. More formally, the experiment where $\mathcal{GG}$ is run on input (m, κ) to get G and then a given probabilistic polynomial time algorithm A is run on input G will result in a collision with negligible probability, for any such A.*
2. *Let $G^m = \{a^m | \ a \in G\}$, and note that G^m is a subgroup of G. Given a uniformly random $g \in G^m$, it is hard to find $h \in G$ with $\phi_p(h) = g$. More formally, the experiment where $\mathcal{GG}$ is run on input (m, κ) to get G, g is sampled at random in G^m, and then a probabilistic polynomial time algorithm A is run on input (G, g), will result in a p'th root of g only with negligible probability, for any such A.*

G can reasonably be conjectured to be commitment friendly if computing the order of G is hard, which can be the case if G is a class group or an RSA group, as we discuss in more detail later. To see this, notice that, if $\phi_p(g) = \phi_p(g')$ and $g \neq g'$, then the order of $g'g^{-1}$ is p, and finding such an element can be conjectured hard if the order of G is not know. Finally, notice that ϕ_p is always collision intractable if $\gcd(p, |G|) = 1$, since in this case ϕ_p is injective.

Commitments from Commitment Friendly Groups. We now construct a single value commitment scheme for $\mathbb{Z}_m$, assuming a group generator algorithm $\mathcal{GG}$ that outputs commitment friendly groups. The construction is described in detail below.

[4] We use G as shorthand for a specification of the group with which you can efficiently choose random elements in G and compute the group operation and inverses.

SV_m: Single-Value Commitment Scheme over $\mathbb{Z}_m$

- *Key generation.* Run $\mathcal{GG}$ on input m and κ to get G. Let $g = a^m$ for a uniformly random $a \in G$. Return $\mathsf{pk} = (G, g)$.
- *Commitment.* Set $\mathcal{R} = G$ and compute $\textsc{Com}_{\mathsf{pk}}(x, r) = g^x r^m$.

Theorem 2. *The construction SV_m from above constitutes a single-value commitment scheme over $\mathbb{Z}_m$.*

Proof. First, observe that the perfect hiding and randomization properties follow immediately from the fact that a commitment to any value is a uniformly random element in G^m.

The homomorphic property follows from

$$\begin{aligned}\textsc{Com}_{\mathsf{pk}}(x, r)\textsc{Com}_{\mathsf{pk}}(x', r') &= g^{x+x'}(rr')^m \\ &= g^{x+x' \bmod m}(g^t rr')^m = \textsc{Com}_{\mathsf{pk}}(x + x' \bmod m, g^t rr')\,,\end{aligned}$$

where t is defined by $x + x' = \big((x + x') \bmod m\big) + tm$. So we can set $R(x, x', r, r') = g^t rr'$. Likewise, we have that $\textsc{Com}_{\mathsf{pk}}(x, r)^{-1} = g^{-x}(r^{-1})^m$, which in turn equals $g^{-x \bmod m}(g^\ell r^{-1})^m$, where ℓ is defined by $-x = (-x \bmod m) + \ell m$, so we set $R(x, -1, r) = g^\ell r^{-1}$. Also, the zero-opening property follows trivially since $\textsc{Com}_{\mathsf{pk}}(x, r)^m = \textsc{Com}_{\mathsf{pk}}(0, \textsc{Com}_{\mathsf{pk}}(x, r))$.

Finally to argue binding, assume an adversary is able to produce $x \neq x'$, r, r' such that $g^x r^m = g^{x'} r'^m$. Setting $s = r' r^{-1}$ we get $g^{x-x'} = s^m$. Since $x - x' \neq 0 \bmod m$, there must be a prime factor p dividing m such that, if p^t is the maximal p-power dividing $x - x'$ and p^k is the maximal power dividing m, we have $p^t < p^k$. The equation above can be written as $(g^{(x-x')/p^t})^{p^t} = (s^{m/p^t})^{p^t}$. Since ϕ_p is assumed collision intractable, we conclude[5] that $g^{(x-x')/p^t} = s^{m/p^t}$. Now, because $p^t < p^k$, we can define $a = s^{m/p^{t+1}}$, and inserting in the equation gives $g^{(x-x')/p^t} = a^p$.

Observe that $\gcd(p, (x - x')/p^t) = 1$ and hence we can compute α, β such that $\alpha p + \beta(x - x')/p^t = 1$. Now set $h = g^\alpha a^\beta$, and observe that

$$h^p = g^{\alpha p}(a^p)^\beta = g^{\alpha p}(g^{(x-x')/p^t})^\beta = g^{\alpha p + \beta(x-x')/p^t} = g\,.$$

Hence, we have found a p'th root of g. This contradicts the assumption that G is commitment friendly, and so the binding property of the commitment scheme holds.

□

[5] We use that if ϕ_p is collision intractable, then it is hard to find $a \neq b$ with $a^{p^t} = b^{p^t}$. Indeed, given such a and b, there must exist $0 \leq i < t$ such that $a^{p^i} \neq b^{p^i}$ but $a^{p^{i+1}} = b^{p^{i+1}}$ which yields the collision (a^{p^i}, b^{p^i}) for ϕ_p.

Examples of Commitment Friendly Groups for Odd m. Theorem 2 shows that a single-value commitment scheme can be obtained from a commitment friendly group, and now we discuss different instantiations of the latter. A first natural example is to choose an RSA modulus N and set $G = \mathbb{Z}_N^*$. If m is odd, we can choose N such that m is relatively prime to $\varphi(N)$. This choice leads $\phi_p(g) = g^p$ to be injective for all $p \mid m$, and hence the collision intractability condition is trivially satisfied. Furthermore, the assumption about p-th roots being hard to compute is essentially the RSA assumption. In more detail, even if m is exponentially large, it can only have a polynomial number of different prime factors, so in contrast to the strong RSA assumption the adversary cannot choose the "public exponent" freely in the p-th root finding experiment, which makes this assumption weaker with respect to the strong RSA assumption.

3.3 Single-Value Commitment Schemes for Even m.

If m is even, we have the problem that collision intractability is violated for $p = 2$ because we have $x^2 = (-x)^2 \bmod N$. As a result, we cannot use the template presented before with $\mathbb{Z}_N^*$ in a direct manner.

If we choose $N = PQ$ such that both P and Q are congruent to 3 modulo 4, then we could use the template with $G = \mathsf{QR}(N)$, the group of quadratic residues modulo N, because in that case this group is of odd order (as shown below), and $\mathsf{QR}(N)$ satisfies the properties of a commitment-friendly group. However, this construction has the practical drawback that it requires an expensive set-up to establish g, because membership in $\mathsf{QR}(N)$ cannot be efficiently decided (so rejection sampling on random elements in $\mathbb{Z}_N^*$ does not work), and the alternative of sampling an element in $\mathbb{Z}_N^*$ and squaring it would require a protocol that keeps the initial value hidden for everybody, only revealing the squared value, which is expensive.

Instead, we will describe a slight variant of the single-value commitment construction from Section 3.2 that solves this problem. First, instead of $\mathbb{Z}_N^*$ or $\mathsf{QR}(N)$, we will use $G = \mathsf{J}^+(N)$, the subgroup of numbers with Jacobi symbol 1 modulo N, and (as above) choose $N = PQ$ such that both P and Q are congruent to 3 modulo 4. With this setup, G has even order (namely $(P-1)(Q-1)/2$), and also $-1 \in G$, so in particular it is unfortunately still the case that, for $x \in G$, $-x$ is also in G and $x^2 = (-x)^2 \bmod n$. To address this issue, we describe below a series of modifications (with respect to our previous construction) that ensure this will not play any effect in the binding property.

The choice of N ensures that the subgroup of quadratic residues $\mathsf{QR}(N)$ has odd order (more precisely, $|\mathsf{QR}(N)| = (P-1)(Q-1)/4$). Therefore we can choose N in such a way that $\gcd(|\mathsf{QR}(N)|, m) = 1$. Furthermore, we have $\mathsf{QR}(N) \leq \mathsf{J}^+(N) \leq \mathbb{Z}_N^*$, where $|\mathsf{J}^+(N)| = 2|\mathsf{QR}(N)|$. As a final background fact, we note that $-1 \in \mathsf{J}^+(N)$ but $-1 \notin \mathsf{QR}(N)$. Also recall that one can compute the Jacobi symbol efficiently given only N, so membership in $\mathsf{J}^+(N)$ can be verified efficiently.

With these tools at hand, we are ready to present our construction of a single-value commitment scheme.

Single-Value Commitment Scheme over $\mathbb{Z}_m$, for even m

- *Key generation.* Return $\mathsf{pk} = (G, g)$, where $G = \mathsf{J}^+(N)$ and N is chosen as above, and $g \leftarrow G$.
- *Commitment.* Set $\mathcal{R} = \{0,1\} \times G$. Given $x \in \mathbb{Z}_m$, choose $(b, r) \in \mathcal{R}$, and output $\textsc{Com}_{\mathsf{pk}}(x, (b, r)) = g^x(-1)^b r^m \bmod N$.

Theorem 3. *Under the assumption that factoring N is hard, the construction SV_m from above constitutes a single-value commitment scheme over $\mathbb{Z}_m$.*

Proof. Perfect hiding follows because r^m is uniform in $\mathsf{QR}(N)$ and therefore $(-1)^b r^m$ is uniform in $\mathsf{J}^+(N)$. The homomorphic and randomization properties are easy to verify in much the same way as in Theorem 2.

For binding, we proceed in a similar way as the aforementioned theorem. If an adversary breaks the binding property this means it would be able to find x, x', r, r', b, b' such that $g^x(-1)^b r^m = g^{x'}(-1)^{b'} r'^m \bmod N$. There must be a prime factor p in m such that the maximal p-power p^t dividing $x - x'$ is smaller than the maximal p-power p^k dividing m. If p is odd, we can proceed in exactly the same way as in Theorem 2, except that in our current case the powers of -1 may lead to the equations being satisfied up to a ± 1 factor. We therefore end up concluding that we can compute h such that $h^p = \pm g \bmod N$. If we have $h^p = -g \bmod N$, then since p is odd, this implies that $(-h)^p = g \bmod N$, so we get a p'th root of g in any case.

The more challenging case is when $p = 2$. In this case, the same arguments will lead to the equation $\left(g^{(x-x')/2^t}\right)^{2^t} = \pm\left(s^{m/2^t}\right)^{2^t} \bmod N$.

First, since both sides are squares and -1 is not a square modulo N, is must be that $\left(g^{(x-x')/2^t}\right)^{2^t} = \left(s^{m/2^t}\right)^{2^t} \bmod N$. Unfortunately, since G has even order, we cannot conclude that $g^{(x-x')/2^t} = s^{m/2^t}$. However, we can instead say that $g^{(x-x')/2^t} = s^{m/2^t}\alpha \bmod N$, where $\alpha^{2^t} \bmod N = 1$.

In particular, α has order a 2-power, and by construction of N, the only possible orders of α would be 1 or 2.

Given the above, one possibility is that α is a non-trivial square root of 1. In this case, we can use α to factor N easily since $(\alpha - 1)(\alpha + 1) = 0 \bmod N$ implies that $\gcd(\alpha - 1, N)$ is either P or Q, which breaks the assumption. Otherwise, α is plus or minus 1. We can now continue the reasoning in the same way as in the original proof, and find that we can compute h such that $h^2 \bmod N = \pm g$. Computing such a square-root easily implies you can factor N and break the computational assumption. □

In both our instantiations for odd and even m, we only need trusted setup to generate the modulus N but not to choose the rest of the public key g. We discuss this further in Section C in the Supplementary Material, where we also discuss instantiations based on class groups.

4 Compressed Σ-Protocol

Let $(\mathcal{G}, \textsc{Com}, R)$ be a vector commitment scheme as defined in Section 3.1, allowing a prover to commit to vectors $\mathbf{x} \in \mathbb{Z}_m^n$. In this section, we consider the problem of proving knowledge of an opening $(\mathbf{x}, \gamma)$ of a commitment $P = \textsc{Com}_{pk}(\mathbf{x}, \gamma)$ satisfying a *linear* constraint $L(\mathbf{x}) = y$ captured by a linear form $L\colon \mathbb{Z}_m^n \to \mathbb{Z}_m$. We construct a *compressed Σ-protocol* [3] for this problem.

In contrast to the compressed Σ-protocols of [3], our protocols are not defined over a finite field $\mathbb{F}$ but over the ring $\mathbb{Z}_m$. Because non-zero challenge differences are required to be invertible, a challenge set $\mathcal{C} \subseteq \mathbb{Z}_m$ has to be *exceptional*. Recall that a subset $\mathcal{C}$ of a ring is said to be exceptional if $c - c'$ is invertible for all distinct $c, c' \in \mathcal{C}$. The largest exceptional subset of $\mathbb{Z}_m$ has cardinality p, where p is the smallest prime divisor of m. Therefore, a straightforward application of [3] can result in (much) smaller challenges sets and therefore larger knowledge errors. In many scenarios, this problem can be overcome by a t-fold parallel repetition reducing the knowledge error from κ down to κ^t [5]. However, as we will see, this parallel repetition approach is sub-optimal and in some cases even insufficient. Namely, since the compression mechanism is 3-special sound, the challenge set is required to have cardinality at least 3. This is impossible when $2 \mid m$. For this reason, we adapt the compressed Σ-protocols of [3] to allow for challenges sampled from an appropriate extension of the ring $\mathbb{Z}_m$.

In Section 4.1, we extend our $\mathbb{Z}_m$-vector commitment scheme to a commitment scheme for vectors defined over an extension $\mathcal{S}$ of the ring $\mathbb{Z}_m$. In Section 4.2, we describe a standard Σ-protocol for proving that a committed vector $\mathbf{x} \in \mathcal{S}^n$ satisfies a linear constraint. This Σ-protocol has a communication complexity that is linear in the dimension n of the secret vector $\mathbf{x} \in \mathcal{S}^n$. Subsequently, we describe a compression mechanism for this Σ-protocol, allowing the communication complexity to be reduced from linear down to logarithmic (Section 4.3). The compressed Σ-protocol described in Section 4.4 is a recursive composition of the basic Σ-protocol and the compression mechanism and has a logarithmic communication complexity for a fixed ring extension $\mathcal{S}$.

4.1 Vector Commitments over Ring Extensions

Let $f(X) \in \mathbb{Z}_m[X]$ be a monic polynomial of degree d and let $\mathcal{S} = \mathbb{Z}_m[X]/(f(X))$ be a degree d ring extension of $\mathbb{Z}_m$. Then the commitment scheme $(\mathcal{G}, \textsc{Com}, R)$ for $\mathbb{Z}_m$-vectors has an immediate extension to a commitment scheme $(\mathcal{G}, \textsc{Com}', R')$ for $\mathcal{S}$-vectors[6] where vectors are committed *coefficient-wise*, i.e.,

$$\textsc{Com}'_{pk}\left(\begin{pmatrix} \sum_{i=1}^{d} a_{1,i} X^{i-1} \\ \vdots \\ \sum_{i=1}^{d} a_{n,i} X^{i-1} \end{pmatrix}, \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_d \end{pmatrix}\right) \mapsto \begin{pmatrix} \textsc{Com}_{pk}\big((a_{1,1}, \ldots, a_{n,1}), \gamma_1\big) \\ \vdots \\ \textsc{Com}_{pk}\big((a_{1,d}, \ldots, a_{n,d}), \gamma_d\big) \end{pmatrix}.$$

This commitment scheme inherits the *homomorphic*, *randomization* and *zero-opening* properties of $(\mathcal{G}, \textsc{Com}, R)$. Furthermore, it has an additional homomorphic property that allows committed vectors to be multiplied by ring elements $a \in \mathcal{S}$. More precisely, for any commitment $c = \textsc{Com}'_{pk}(\mathbf{x}, \gamma)$ and $a \in \mathcal{S}$, the commitment c^a is well-defined and can be opened to $a \cdot \mathbf{x} \in \mathcal{S}^n$. To see this note that any $a \in \mathcal{S}$ corresponds to a matrix $\mathcal{M}(a) \in \mathbb{Z}_m^{d \times d}$, such that for all $b \in \mathcal{S}$:

$$a \cdot \sum_{i=1}^{d} b_i X^{i-1} = \sum_{i=1}^{d} c_i X^{i-1} \in \mathcal{S} \quad \Longleftrightarrow \quad \mathcal{M}(a) \begin{pmatrix} b_1 \\ \vdots \\ b_d \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_d \end{pmatrix} \in \mathbb{Z}_m^d .$$

By lifting this matrix to $\mathbb{Z}^{d \times d}$,[7] it follows that the homomorphic operation c^a can be expressed in terms of the standard homomorphic properties of

[6] Notice that in Section 2.1 we only defined commitments for vectors over $\mathbb{Z}_m$, while here we need commitments for vectors over $\mathcal{S}$, and moreover, we need these to be homomorphic as a $\mathcal{S}$-module. This notion is defined in a similar manner as the homomorphism from Section 2.1.

[7] We lift to $\mathbb{Z}^{d \times d}$ because the homomorphic properties are defined over $\mathbb{Z}$.

the $\mathbb{Z}_m$-commitment scheme $(\mathcal{G}, \textsc{Com}, R)$. As before, we write $R'(\mathbf{x}, a, \gamma)$ for the randomness required to open c^a to $a \cdot \mathbf{x} \in \mathcal{S}^n$. We say that this commitment scheme is $\mathcal{S}$-homomorphic. Finally, a $\mathbb{Z}_m$-vector commitment P can also be viewed as a $\mathcal{S}$-vector commitment $(P, 1, \ldots, 1)$, now with $\mathcal{S}$-homomorphic properties.

Remark 1. Alternatively, one can commit to a vector

$$(\sum_{i=1}^{d} a_{1,i} X^{i-1}, \ldots, \sum_{i=1}^{d} a_{n,i} X^{i-1}) \in \mathcal{S}^n$$

by committing to all coefficients $(a_{1,1}, \ldots, a_{n,d}) \in \mathbb{Z}_m^{nd}$ in a single $\mathbb{Z}_m$-vector commitment. This approach results in commitments that are a factor d smaller. However, these commitments are only $\mathbb{Z}_m$-homomorphic. Hence, to obtain a scheme that is $\mathcal{S}$-homomorphic, it is crucial that the commitment function $\textsc{Com}'_{pk}$ is a coefficient-wise application of the $\mathbb{Z}_m$-commitment function $\textsc{Com}_{pk}$.

4.2 Standard Σ-Protocol

The reason for considering vectors defined over the ring extension $\mathcal{S} = \mathbb{Z}_m[X]/(f(X))$ is that when this extension is appropriately chosen it contains larger *exceptional subsets* than the ring $\mathbb{Z}_m$. Namely, if $f(X)$ is irreducible modulo all prime divisors of m, then $\mathcal{S}$ contains an exceptional subset of cardinality p^d where p is the smallest prime dividing m. This allows us to design (compressed) Σ-protocols with larger challenge sets and therefore smaller knowledge errors. From now on we will assume $f(X)$ to be of this form and $\mathcal{C} \subseteq \mathcal{S}$ to be an exceptional subset of cardinality p^d.

Protocol 1, denoted by Π_1, is a standard Σ-protocol, with challenge set $\mathcal{C}$, for proving knowledge of a commitment opening satisfying a linear constraint, i.e., it is a Σ-protocol for relation

$$\mathcal{X}^d = \{(P, y; \mathbf{x}, \gamma) : \textsc{Com}'_{pk}(\mathbf{x}, \gamma) = P,\ L(\mathbf{x}) = y\},$$

where $\mathbf{x} \in \mathcal{S}^n$ and $L \colon \mathcal{S}^n \to \mathcal{S}$ is a linear form. The properties of Π_1 are summarized in Theorem 4.

This Σ-protocol can also instantiated for relation $\mathcal{X}$. More precisely, to prove knowledge of an opening $(\mathbf{x}; \gamma) \in \mathbb{Z}_m^n \times \mathcal{R}$ to the $\mathbb{Z}_m$-vector commitment P satisfying the $\mathbb{Z}_m$-linear constraint $L(\mathbf{x}) = y$, Π_1 can be instantiated with statement $\big((P, 1, \ldots, 1), y\big)$ for relation $\mathcal{X}^d$. The same holds for the protocols in the subsequent sections.

Protocol 1 Standard Σ-Protocol Π_1 for relation $\mathcal{X}^d$.

$\textsc{Input}(P, y; \mathbf{x}, \gamma)$
$P = \textsc{Com}'_{pk}(\mathbf{x}, \gamma)$
$y = L(\mathbf{x})$

Prover		Verifier
$\mathbf{r} \leftarrow_R \mathcal{S}^n$		
$\rho \leftarrow_R \mathcal{R}^d$		
$A = \textsc{Com}'_{pk}(\mathbf{r}, \rho)$		
$t = L(\mathbf{r})$	$\xrightarrow{A,t}$	
		$c \leftarrow_R \mathcal{C} \subseteq \mathcal{S}$
	$\xleftarrow{c}$	
$\mathbf{z} = \mathbf{r} + c\mathbf{x}$		
$\psi = R'\big(\mathbf{r}, c\mathbf{x}, \rho, R'(\mathbf{x}, c, \gamma)\big)$		
	$\xrightarrow{\mathbf{z},\psi}$	$\textsc{Com}'_{pk}(\mathbf{z}, \psi) \stackrel{?}{=} A \cdot P^c$
		$L(\mathbf{z}) \stackrel{?}{=} t + cy$

Theorem 4 (Standard Σ-Protocol). *Protocol Π_1 (as defined in Protocol 1) is a Σ-protocol for relation $\mathcal{X}^d$. More precisely, it is a 3-round protocol that is perfectly complete, special honest-verifier zero-knowledge and unconditionally knowledge sound with knowledge error $1/p^d$, where p is the smallest prime dividing m.*

Proof. **Completeness** follows directly by the homomorphic properties of $\textsc{Com}_{pk}(\cdot)$ and the linearity of L.

SHVZK: We simulate a transcript as follows. Given a challenge c, sample $(\mathbf{z}, \psi) \leftarrow_R \mathcal{S}^n \times \mathcal{R}^d$ uniformly at random and let $A = \textsc{Com}'_{pk}(\mathbf{z}, \psi) \cdot P^{-c}$ and $t = L(\mathbf{z}) - cy$. By the randomization property of $\textsc{Com}'_{pk}$ it follows that the simulated transcripts $(A, t, c, \mathbf{z}, \psi)$ have exactly the same distribution as honestly generated transcripts.

Knowledge Soundness: We show that Π_1 is special-sound. Knowledge soundness is then implied. Let $(A, t, c, \mathbf{z}, \psi)$, $(A, t, c', \mathbf{z}', \psi')$ be two accepting transcripts with $c \neq c' \in \mathcal{C}$, and let $\tilde{c} = (c - c')^{-1}$. Then define

$$\tilde{\mathbf{z}} := \tilde{c}(\mathbf{z} - \mathbf{z}'),$$
$$\tilde{\psi} := R'\big(\tilde{c}\mathbf{z}, -\tilde{c}\mathbf{z}', R'(\mathbf{z}, \tilde{c}, \psi), R'(\mathbf{z}', -\tilde{c}, \psi')\big).$$

By the homomorphic properties of $\textsc{Com}'_{pk}(\cdot)$ and since the transcripts are accepting, it follows that $\textsc{Com}'_{pk}(\tilde{\mathbf{z}}, \tilde{\psi}) = P^{\tilde{c}(c-c')} = P \cdot P^{\ell m}$ for some $\ell \in \mathbb{Z}$. Hence, by the zero-opening property of $\textsc{Com}'_{pk}(\cdot)$, $(\tilde{\mathbf{z}}, \bar{\psi})$

is an opening of commitment P, where $\bar{\psi} = R'(\tilde{\mathbf{z}}, 0, \tilde{\psi}, R'(P^{-\ell}))$. By the linearity of L, it additionally follows that $L(\tilde{\mathbf{z}}) = y$, i.e., $(\tilde{\mathbf{z}}, \bar{\psi})$ is a witness for statement $(P, y) \in L_{\mathcal{X}^d}$.

□

Remark 2. The above Σ-protocol can be used to prove knowledge of the openings of d different $\mathbb{Z}_m$-commitments $P_1, \ldots, P_d$ by defining the $\mathcal{S}$-commitment $P = (P_1, \ldots, P_d)$, i.e., a protocol for proving knowledge of d witnesses for relation $\mathcal{X}^1$. The naive approach for achieving this functionality would be to instantiate d different Σ-protocols defined directly over $\mathbb{Z}_m$. However, as displayed in Table 1 this results in a larger knowledge error. Alternatively, one could apply standard amortization techniques to prove knowledge of d witnesses with the same communication costs as proving knowledge of only 1 witness (see for example [3]). This approach reduces the communication costs by a factor d. However, it comes at the cost of increasing the knowledge error.

Table 1. Properties of different Σ-protocols for proving knowledge of d witnesses for relation $\mathcal{X}^1$. Columns 2-4 contain communication costs, while the last column contains knowledge error.

Protocol	# $\mathbb{Z}_m$-elements	# $\mathcal{R}$-elements	# $\mathbb{Z}_m^n$-Commitments	K. error
d Separate Σ-Protocols	$d(n+1)$	d	d	$1/p$
Amortized Σ-Protocol	$n+1$	1	1	d/p
Our Σ-Protocol Π_1	$d(n+1)$	d	d	$1/p^d$

4.3 Compression Mechanism

The communication complexity of the standard Σ-protocol Π_1 is linear in the dimension n of the secret input vector $\mathbf{x} \in \mathcal{S}^n$. The compression mechanism for Σ-protocols of [3], based on Bulletproof's folding technique [15,17], allows the communication complexity to be reduced from linear down to logarithmic. A key observation of this compression mechanism is that the final message of protocol Π_1 is a witness for relation $\mathcal{X}^d$, i.e., the final message is a trivial proof-of-knowledge (PoK) for this relation. Therefore, this message can also be replaced by another PoK for relation $\mathcal{X}^d$. In particular, it can be replaced by a PoK with a smaller communication complexity. Compression mechanism Π_2, described in Protocol 2, is a PoK for relation $\mathcal{X}^d$. Bulletproof's folding technique takes an

n-dimensional witness $\mathbf{x} = (\mathbf{x}_L, \mathbf{x}_R) \in \mathcal{S}^n$ and, given a challenge $c \in \mathcal{C}$, it folds the left and right halves $\mathbf{x}_L, \mathbf{x}_R \in \mathcal{S}^{n/2}$ onto each other obtaining a new message $\mathbf{z} = \mathbf{x}_L + c\mathbf{x}_R$ of dimension $n/2$. This technique reduces the communication complexity by roughly a factor 2. The properties of this protocol are summarized in Theorem 5. For more details we refer to [3].

Protocol 2 Compression Mechanism Π_2 for Relation $\mathcal{X}^d$.

$\textsc{Input}(P, y; \mathbf{x}, \gamma)$
$P = \textsc{Com}'_{pk}(\mathbf{x}, \gamma)$
$y = L(\mathbf{x})$

Prover		Verifier
$\rho \leftarrow_R \mathcal{R}^d$ $A = \textsc{Com}'_{pk}((\mathbf{0}, \mathbf{x}_L), \rho),\ a = L(\mathbf{0}, \mathbf{x}_L)$ $B = \textsc{Com}'_{pk}((\mathbf{x}_R, \mathbf{0}), \rho),\ b = L(\mathbf{x}_R, \mathbf{0})$	$\xrightarrow{A,B,a,b}$	
		$c \leftarrow_R \mathcal{C} \subseteq \mathcal{S}$
	$\xleftarrow{c}$	
$\mathbf{z} = \mathbf{x}_L + c\mathbf{x}_R$ $\psi_1 = R'\big((\mathbf{0}, \mathbf{x}_L), c\mathbf{x}, \rho, R'(\mathbf{x}, c, \gamma)\big)$ $\psi_2 = R'\big((\mathbf{x}_R, \mathbf{0}), c^2, \rho\big)$ $\psi = R'\big((\mathbf{0}, \mathbf{x}_L) + c\mathbf{x}, c^2(\mathbf{x}_R, \mathbf{0}), \psi_1, \psi_2\big)$		
	$\xrightarrow{\mathbf{z},\psi}$	$\textsc{Com}'_{pk}\big((c\mathbf{z}, \mathbf{z}), \psi\big) \stackrel{?}{=} A \cdot P^c \cdot B^{c^2}$ $L(c\mathbf{z}, \mathbf{z}) \stackrel{?}{=} a + cy + c^2 b$

Theorem 5 (Compression Mechanism). *Let n be even. Protocol Π_2 (as defined in Protocol 2) is a 3-round protocol for relation $\mathcal{X}^d$. It is perfectly complete and unconditionally knowledge sound with knowledge error $2/p^d$, where p is the smallest prime diving m. Its communication costs are*

- *From Prover to Verifier:* 2 $\mathcal{S}$*-commitments,* $n/2+2$ *elements in* $\mathcal{S}$ *and* 1 *elements in* $\mathcal{R}^d$.
- *From Verifier to Prover:* 1 *challenge in* $\mathcal{C} \subseteq \mathcal{S}$.

Due to space limitations, the proof of this Theorem is presented in Section D in the Supplementary Material.

4.4 Compressed Σ-Protocol

To reduce the communication costs of the Σ-protocol Π_1 down to logarithmic the compression mechanism is applied recursively, i.e., instead of

sending the final message of protocol Π_2 the protocol is applied again until the dimension of the final message equals 4. Note that the compression mechanism could be applied even further, reducing the dimension of the final message to 2 or 1. However, since the prover has to send 4 elements in every compression, this would result in a sub-optimal communication costs. This recursive composition is referred to a *Compressed Σ-Protocol*, it is denoted by

$$\Pi_c = \underbrace{\Pi_2 \diamond \cdots \diamond \Pi_2}_{\lceil \log_2(n) \rceil - 2 \text{ times}} \diamond \Pi_1 \,,$$

and its properties are summarized in the following theorem. In particular the protocol is $(2, 3, \ldots, 3)$-special-sound, which has recently been shown to tightly imply knowledge soundness [4].

Theorem 6 (Compressed Σ-Protocol). *Let $n = 2^\mu \geq 4$. Then, Protocol Π_c is a $(2\mu - 1)$-round protocol for relation $\mathcal{X}^d$. It is perfectly complete, special honest-verifier zero-knowledge. Moreover, it is unconditionally $(2, 3, \ldots, 3)$-special-sound and therefore knowledge sound with knowledge error*

$$1 - \frac{(p^d - 1)(p^d - 2)^{\mu-2}}{p^{d(\mu-1)}} \leq \frac{2\mu - 3}{p^d} \,,$$

where p is the smallest prime dividing m. Its communication costs are

- *From Prover to Verifier: $2\mu - 3$ $\mathcal{S}$-commitments, $2\mu + 1$ elements in $\mathcal{S}$ and 1 element in $\mathcal{R}^d$.*
- *From Verifier to Prover: $\mu - 1$ challenges in $\mathcal{C} \subseteq \mathcal{S}$.*

In practical applications, Π_c should be instantiated with knowledge error at most $2^{-\lambda}$, where λ denotes the security parameter. To this end, we choose a ring extension $\mathcal{S}$ of degree

$$d \geq \frac{1 + \lambda + \log\log n}{\log p} = \mathcal{O}(\lambda + \log\log n) \,.$$

Hence, to obtain a knowledge error negligible in the security parameter, the degree must depend on the input dimension n. However, thus far we have only consider the communication complexity for *fixed* ring extensions $\mathcal{S}$ of degree d and thus with *fixed*, not necessarily negligible, knowledge error. In fact, the communication complexity of Π_c is only logarithmic in n for fixed $\mathcal{S}$ and d. Taking into account that in practice $d = \mathcal{O}(\lambda + \log\log n)$, shows that the communication complexity is actually

$\mathcal{O}(\lambda \log n + \log n \log\log n)$, i.e., it is not logarithmic in n. However, this is still an improvement over the polylogarithmic communication complexity achieved by the naive approach using integer commitment schemes.

Further, the knowledge error of the Compressed Σ-Protocol Π_c shows that we must choose the degree d of the ring extension such that $p^d > 2$. In particular, if $p = 2$ the compression mechanism can not be defined directly over $\mathbb{Z}_m$. If $p > 2$, then the compressed Σ-protocol could have been defined over $\mathbb{Z}_m$ directly. However, this would result in a larger knowledge error. Reducing this knowledge error by a d-fold parallel composition would result in exactly the same communication costs as the protocol defined over the ring extension $\mathcal{S}$. However, this parallel composition approach results in a knowledge error of

$$\frac{\left(p^{\mu-1} - (p-1)(p-2)^{\mu-2}\right)^d}{p^{d(\mu-1)}} \leq \frac{(2\mu-3)^d}{p^d},$$

which is larger than the the knowledge error of our protocol. Hence, even for the case $p > 2$, it is beneficial to define the protocols over the ring extension $\mathcal{S}$. Moreover, this approach allows a prover to prove d $\mathbb{Z}_m$-statements simultaneously (coordinate-wise) with exactly the same costs as proving only 1 statement.

Remark 3. The communication complexity of protocol Π_c can be further reduced with roughly factor 1/2, by incorporating the linear form evaluation $L(\mathbf{x})$ into the commitment. More precisely, before evaluating the Compressed Σ-Protocol, the verifier sends a random challenge $c \in \mathcal{C} \subseteq \mathcal{S}$ to the prover, and relation $\mathcal{X}$ is transformed into relation

$$\mathcal{X}_c^d = \{(P, y; \mathbf{x}, \gamma) : \textsc{Com}'_{pk}(\mathbf{x}, c \cdot L(\mathbf{x}), \gamma) = P\}.$$

After this transformation the prover does not have to send the linear form evaluations a, b in compression mechanism Π_2 to the verifier. For more details see [3].

Remark 4. With small adaptations to existing work, we can use our Σ-protocols to prove non-linear constraints. Namely, following [3], we can "linearize" this type of constraints by an arithmetic secret sharing based technique, after which the protocols described in previous sections can be used in a black-box manner. In the lattice-based compressed Σ-protocols of [4] it was already shown how to adapt this techniques to the ring scenario.[8] For a general arithmetic circuit C over $\mathcal{S}$ we can then construct

[8] On the other hand, since they only considered rings with large enough exceptional sets, their protocol for proving linear statements could be defined over the base ring and therefore the adaptations of the previous sections were not required in [4].

a protocol that can prove the relation

$$\{(P, y; \mathbf{x}, \gamma) : \textsc{Com}'_{pk}(\mathbf{x}, \gamma) = P,\ C(\mathbf{x}) = y\},$$

with communication complexity logarithmic in the dimension n of $\mathbf{x} \in \mathcal{S}^n$ and the number of multiplication gates m in the circuit C.

Finally, our protocols are also compatible with the Fiat-Shamir heuristic. We discuss this in Section E in the Supplementary Material.

5 An Application: Verifiable Computation with Context-Hiding

In this section, we argue that our commitments and compressed Σ-protocols over rings are useful in the context of proofs of correct computation on homomorphically encrypted data. We illustrate this concretely by considering the problem of verifiable computation on encrypted data supporting non-deterministic computations and context hiding from the recent work [14].

In verifiable computation [31], a client wants to delegate a (typically expensive) computation $\mathbf{y} = g(\mathbf{x})$ to a server, which must later prove that the computation has been carried out correctly. When the client does not want the server to learn information about the actual inputs $\mathbf{x}$ of the computation, we speak of private verifiable computation. To address this privacy consideration, several works [28,29,14] have proposed to combine verifiable computation and homomorphic encryption: the client encrypts the input data with a fully homomorphic scheme and sends the ciphertexts $\mathsf{ct}_{x_1}, \mathsf{ct}_{x_2}, \dots$ to the server, which carries out the corresponding computation $\hat{g}(\mathsf{ct}_{x_1}, \mathsf{ct}_{x_2}, \dots)$ on the encrypted data and proves its correctness via a verifiable computation scheme. One problem of this approach is how to reconcile the ciphertext space of homomorphic encryption (which can generally be a polynomial ring of the form $(\mathbb{Z}_q[X]/(f))^D$ where q is not necessarily prime) and the space on which the verifiable computation system operates (typically a large field).

This motivated [14] to introduce a scheme that provides flexibility in this combination, where the idea is to "homomorphically hash" the computation on the ciphertext space into a computation on a smaller Galois ring $\mathbb{Z}_q[X]/(h)$ (where $d_h := \deg h \ll d_f := \deg f$)[9], and then

[9] More precisely, the computation is first lifted from $\mathbb{Z}_q[X]/(f)$ to $\mathbb{Z}_q[X]$ and then mapped to $\mathbb{Z}_q[X]/(h)$ via the canonical homomorphism (which is a ring homomorphism), for an irreducible $h \in \mathbb{Z}_q[X]$ chosen at random by the verifier among all such h of a given degree. Moreover, the technique needs that q is a prime power, but one can easily reduce to this case via the Chinese Remainder Theorem.

to prove the correctness of the computation *on the hash images* of the ciphertexts, for which one can use as succinct argument a version of the GKR protocol [34] adapted for Galois rings, presented in [21].

However, an additional challenge appears if the privacy of the input data needs to be preserved with respect to a (public) verifier too. In this case, we speak of verifiable computation with context-hiding (as introduced in [29]) and [14] proposes to use the following commit-and-proof strategy: the client commits publicly to the ciphertexts ct_x sent to the server, which blinds the resulting output ciphertexts with encryptions ct_0 of 0, commits to them and sends the resulting ciphertexts ct_y to the verifier. After receiving the hash function H, the server publishes[10] the hash images $H(\mathsf{ct}_x)$, $H(\mathsf{ct}_0)$ of all input ciphertexts and encryptions of zero, and carries out a commit-and-proof argument that these have been computed correctly. This strategy even extends to non-deterministic computations $\mathbf{y} = g(\mathbf{x}; \mathbf{w})$ which may depend on randomness $\mathbf{w}$ chosen by the server (which we want to be hidden even if the verifier is the client who provided the data); in this case the server also commits to the encryptions ct_w of these random values and publishes the hashes.

The strategy above requires the server to prove that for every involved ciphertext ct, $H(\mathsf{ct})$ has been correctly computed, in addition to the fact that $\mathsf{ct}_0, \mathsf{ct}_w$ are correct encryptions. While [14] propose these generic solutions, they leave as an open question the existence of succinct commit-and-proof arguments that directly handle statements over (Galois) rings, so that there is no need to emulate the ring arithmetic with an argument over a finite field, which causes considerable overhead.

Our homomorphic commitments and compressed Sigma-protocols are well suited for this context: indeed, the hash functions are $\mathbb{Z}_q$-linear maps $H : \mathbb{Z}_q^{2d_f} \to \mathbb{Z}_q[X]/(h)$.[11] In fact, by the right amortization technique, one can even turn the hash statements for d_h ciphertexts into one $\mathcal{S}$-form $H : \mathcal{S}^{2d_f} \to \mathcal{S}$ where $\mathcal{S} = \mathbb{Z}_q[X]/(h)$. On the other hand, proving the correctness of the encryptions can be reduced to range proofs, which can be addressed by adapting the efficient protocols for range proofs described in [3] to a large enough extension ring of $\mathbb{Z}_q$.

[10] We are describing one of the versions in [14] where these hashes can be published without harm to privacy. A more general version of the protocol would require to commit to the hashes too, and prove the correctness of computation of $\hat{g}$ via a commit-and-proof argument rather than GKR, which can also be done with compressed Σ-protocols (see Section E.1) but is considerably more complex.

[11] This is assuming that the BV scheme [16] has been used to encrypt (as [14] does for concrete instantiations), which produces a ciphertext consisting of two polynomials of degree at most $d_f - 1$, hence the $2d_f$.

This provides a simple and efficient way of instantiating the type of commit-and-proof arguments left open in [14]. In particular, the communication complexity of the commit-and-proof part of the protocol becomes $O(\log d_f \cdot M)$, where M is the size of the inputs and outputs of the computation (the total size of all x, w and y). The constant hidden in the O-notation depends on the commitment instantiation, and the noise parameters of the encryption scheme (because of the range proofs), the latter of which depends on the complexity of the target computation g.

Acknowledgments

This paper was prepared in part for information purposes by the Artificial Intelligence Research group of JPMorgan Chase & Co and its affiliates ("JP Morgan"), and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

References

1. M. Abspoel, R. Cramer, I. Damgård, D. Escudero, and C. Yuan. Efficient information-theoretic secure multiparty computation over $\mathbb{Z}/p^k\mathbb{Z}$ via galois rings. In D. Hofheinz and A. Rosen, editors, *TCC 2019, Part I*, volume 11891 of *LNCS*, pages 471–501. Springer, Heidelberg, Dec. 2019.
2. S. Ames, C. Hazay, Y. Ishai, and M. Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 2017*, pages 2087–2104. ACM Press, Oct. / Nov. 2017.
3. T. Attema and R. Cramer. Compressed Σ-protocol theory and practical application to plug & play secure algorithmics. In D. Micciancio and T. Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 513–543. Springer, Heidelberg, Aug. 2020.
4. T. Attema, R. Cramer, and L. Kohl. A compressed Σ-protocol theory for lattices. In T. Malkin and C. Peikert, editors, *CRYPTO 2021, Part II*, volume 12826 of *LNCS*, pages 549–579, Virtual Event, Aug. 2021. Springer, Heidelberg.

5. T. Attema and S. Fehr. Parallel repetition of $(k_1, \ldots, k_\mu)$-special-sound multi-round interactive proofs. Cryptology ePrint Archive, Report 2021/1259, 2021. https://eprint.iacr.org/2021/1259.
6. T. Attema, S. Fehr, and M. Klooß. Fiat-shamir transformation of multi-round interactive proofs. Cryptology ePrint Archive, Report 2021/1377, 2021. https://eprint.iacr.org/2021/1377.
7. C. Baum, L. Braun, A. Munch-Hansen, and P. Scholl. Appenzeller to brie: Efficient zero-knowledge proofs for mixed-mode arithmetic and z2k. 2021.
8. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In D. Catalano and R. De Prisco, editors, *SCN 18*, volume 11035 of *LNCS*, pages 368–385. Springer, Heidelberg, Sept. 2018.
9. E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society Press, May 2014.
10. E. Ben-Sasson, A. Chiesa, M. Riabzev, N. Spooner, M. Virza, and N. P. Ward. Aurora: Transparent succinct arguments for R1CS. In Y. Ishai and V. Rijmen, editors, *EUROCRYPT 2019, Part I*, volume 11476 of *LNCS*, pages 103–128. Springer, Heidelberg, May 2019.
11. R. Bendlin, I. Damgård, C. Orlandi, and S. Zakarias. Semi-homomorphic encryption and multiparty computation. In K. G. Paterson, editor, *EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 169–188. Springer, Heidelberg, May 2011.
12. F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In G. Pernul, P. Y. A. Ryan, and E. R. Weippl, editors, *ESORICS 2015, Part I*, volume 9326 of *LNCS*, pages 305–325. Springer, Heidelberg, Sept. 2015.
13. A. R. Block, J. Holmgren, A. Rosen, R. D. Rothblum, and P. Soni. Time- and space-efficient arguments from groups of unknown order. In T. Malkin and C. Peikert, editors, *CRYPTO 2021, Part IV*, volume 12828 of *LNCS*, pages 123–152, Virtual Event, Aug. 2021. Springer, Heidelberg.
14. A. Bois, I. Cascudo, D. Fiore, and D. Kim. Flexible and efficient verifiable computation on encrypted data. In J. Garay, editor, *PKC 2021, Part II*, volume 12711 of *LNCS*, pages 528–558. Springer, Heidelberg, May 2021.
15. J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 327–357. Springer, Heidelberg, May 2016.
16. Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-LWE and security for key dependent messages. In P. Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 505–524. Springer, Heidelberg, Aug. 2011.
17. 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.
18. B. Bünz, B. Fisch, and A. Szepieniec. Transparent SNARKs from DARK compilers. In A. Canteaut and Y. Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 677–706. Springer, Heidelberg, May 2020.
19. D. Catalano, M. Di Raimondo, D. Fiore, and I. Giacomelli. Mon$\mathbb{Z}_{2^k}$a: Fast maliciously secure two party computation on $\mathbb{Z}_{2^k}$. In A. Kiayias, M. Kohlweiss,

P. Wallden, and V. Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 357–386. Springer, Heidelberg, May 2020.
20. M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, and G. Zaverucha. Post-quantum zero-knowledge and signatures from symmetric-key primitives. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 2017*, pages 1825–1842. ACM Press, Oct. / Nov. 2017.
21. S. Chen, J. H. Cheon, D. Kim, and D. Park. Verifiable computing for approximate computation. Cryptology ePrint Archive, Report 2019/762, 2019. https://eprint.iacr.org/2019/762.
22. R. Cramer. Modular design of secure yet practical cryptographic protocols. *Ph. D. Thesis, CWI and University of Amsterdam*, 1996.
23. R. Cramer and I. Damgård. Zero-knowledge proofs for finite field arithmetic; or: Can zero-knowledge be for free? In H. Krawczyk, editor, *CRYPTO'98*, volume 1462 of *LNCS*, pages 424–441. Springer, Heidelberg, Aug. 1998.
24. R. Cramer, I. Damgård, D. Escudero, P. Scholl, and C. Xing. SPD $\mathbb{Z}_{2^k}$: Efficient MPC mod 2^k for dishonest majority. In H. Shacham and A. Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 769–798. Springer, Heidelberg, Aug. 2018.
25. R. Cramer, I. Damgård, and B. Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In Y. Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 174–187. Springer, Heidelberg, Aug. 1994.
26. I. Damgård, D. Escudero, T. K. Frederiksen, M. Keller, P. Scholl, and N. Volgushev. New primitives for actively-secure MPC over rings with applications to private machine learning. In *2019 IEEE Symposium on Security and Privacy*, pages 1102–1120. IEEE Computer Society Press, May 2019.
27. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985.
28. D. Fiore, R. Gennaro, and V. Pastro. Efficiently verifiable computation on encrypted data. In G.-J. Ahn, M. Yung, and N. Li, editors, *ACM CCS 2014*, pages 844–855. ACM Press, Nov. 2014.
29. D. Fiore, A. Nitulescu, and D. Pointcheval. Boosting verifiable computation on encrypted data. In A. Kiayias, M. Kohlweiss, P. Wallden, and V. Zikas, editors, *PKC 2020, Part II*, volume 12111 of *LNCS*, pages 124–154. Springer, Heidelberg, May 2020.
30. C. Ganesh, A. Nitulescu, and E. Soria-Vazquez. Rinocchio: Snarks for ring arithmetic. *IACR Cryptol. ePrint Arch.*, 2021:322, 2021.
31. R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 465–482. Springer, Heidelberg, Aug. 2010.
32. I. Giacomelli, J. Madsen, and C. Orlandi. ZKBoo: Faster zero-knowledge for Boolean circuits. In T. Holz and S. Savage, editors, *USENIX Security 2016*, pages 1069–1083. USENIX Association, Aug. 2016.
33. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In A. Aho, editor, *19th ACM STOC*, pages 218–229. ACM Press, May 1987.
34. S. Goldwasser, Y. T. Kalai, and G. N. Rothblum. Delegating computation: interactive proofs for muggles. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 113–122. ACM Press, May 2008.
35. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985.

36. J. Groth. On the size of pairing-based non-interactive arguments. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016.
37. M. Joye and B. Libert. Efficient cryptosystems from 2^k-th power residue symbols. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 76–92. Springer, Heidelberg, May 2013.
38. T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In K. Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 308–318. Springer, Heidelberg, May / June 1998.
39. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, Heidelberg, May 1999.
40. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In J. Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 129–140. Springer, Heidelberg, Aug. 1992.
41. C.-P. Schnorr. Efficient identification and signatures for smart cards. In G. Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, Aug. 1990.
42. R. S. Wahby, I. Tzialla, a. shelat, J. Thaler, and M. Walfish. Doubly-efficient zkSNARKs without trusted setup. In *2018 IEEE Symposium on Security and Privacy*, pages 926–943. IEEE Computer Society Press, May 2018.
43. D. Wikström. Special soundness in the random oracle model. Cryptology ePrint Archive, Report 2021/1265, 2021. https://eprint.iacr.org/2021/1265.
44. T. Xie, J. Zhang, Y. Zhang, C. Papamanthou, and D. Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 733–764. Springer, Heidelberg, Aug. 2019.

Supplementary Material

A Technical Overview (Extended)

As a starting point, we begin with the theory of compressed Σ-protocols presented in [3], and analyze in detail which parts are inherently dependent on the underlying algebraic structure being $\mathbb{Z}_p$ for a prime number p. Let us begin with a short overview of the techniques in [3], which will be followed by the aforementioned analysis.

Overview of the Techniques in [3]. The basic "pivot" presented in [3], from which most of their results are derived, is a Σ-protocol that enables a prover to convince a verifier that, given a commitment and certain value, he knows how to open that commitment to a vector that maps, under a some public linear mapping, to the given value. More precisely, let $\mathbb{G}$ be a finite abelian group of prime order q. Let P be a *Pedersen commitment* $P = h^{\gamma} \prod_{i=1}^{n} g_i^{x_i}$ to a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_q^n$, where the $g_1, \ldots, g_n, h$ are uniformly random elements from $\mathbb{G}$ sampled in a setup phase. Also, let $L : \mathbb{Z}_q^n \to \mathbb{Z}_q$ be a linear form, and let $y \in \mathbb{Z}_q$ be a given value. The authors of [3] devise a communication efficient Σ-protocol that enables a prover to prove knowledge of $\mathbf{x}$, the vector underlying the commitment P, while proving that this vector satisfies $L(\mathbf{x}) = y$. At a high level, such protocol is achieved by first considering a basic and natural three-move Σ-protocol for this relation, which would involve the prover sending a long response to the challenge provided by the verifier, and then optimizing this last step by making use of a more efficient proof of knowledge of this response, which is derived from the techniques in Bulletproofs [15,17].

The basic three-move Σ-protocol looks as follows:

1. The prover samples $\mathbf{r} \leftarrow \mathbb{Z}_q^n$ and $\rho \leftarrow \mathbb{Z}_q$, and sends $t = L(\mathbf{r})$ and $A = h^{\rho} \prod_{i=1}^{n} g_i^{r_i}$ to the verifier;
2. The verifier samples a challenge $c \leftarrow \mathbb{Z}_q$ to the prover;
3. The prover responds with $\mathbf{z} = c\mathbf{x} + \mathbf{r}$ and $\phi = c\gamma + \rho$, and the verifier checks that $h^{\phi} \prod_{i=1}^{n} g_i^{z_i} = AP^c$ and $L(\mathbf{z}) = cy + t$.

In the second part, instead of the prover sending $\mathbf{z}$ and ϕ as the last step of the protocol above, the prover uses a more efficient proof of knowledge to prove to the verifier that he knows $\mathbf{z}$ and ϕ satisfying $h^{\phi} \prod_{i=1}^{n} g_i^{z_i} = AP^c$ and $L(\mathbf{z}) = cy + t$. This proof has logarithmic (in n)

communication complexity, and it is based on the core pivot of the Bulletproof protocol [15,17]. It is quite difficult to provide a general intuition on these techniques in a few paragraphs but, in a nutshell, they consist of splitting the data into two halves, and combining them via a new challenge that makes it hard for the prover to cheat. This can be recursed to obtain logarithmic communication.

Dependencies on $\mathbb{Z}_q$ for a Prime q. At this point, we can identify two main locations in the protocol from [3] that seem to depend heavily on the algebraic structure being $\mathbb{Z}_q$ for a prime q.

- *Challenges and soundness.* To ensure low cheating probability, challenges are sampled by the verifier to somehow "randomize" the response the prover needs to provide. Ultimately, to show special soundness, one must show that successfully replying to multiple challenges enables us to extract a witness. This is typically done by solving a linear equation, or more generally, a set of linear equations. Such approach proves difficult when not operating over a field given the lack of invertible elements.
- *Homomorphic commitments.* The techniques from [3] depend on a commitment scheme that is homomorphic over the desired algebraic structure. We considered above Pedersen commitments, but the results from [3] include other constructions whose security depends on different assumptions such as Strong RSA and Knowledge-of-Exponent, and Lattices were considered in [4]. All of these techniques, however, require a specific type of modulus. For instance, Pedersen commitments are defined over cyclic groups, and the construction from [3] based on the Strong RSA assumption only allows for RSA moduli.

Our Approach to Extend to $\mathbb{Z}_m$ for any m.

- *Challenges and soundness.* Fortunately, we can address the issue of soundness and non-invertibility by sampling challenges from an *exceptional set*, which consists of elements whose non-zero pairwise differences are invertible. This approach has been used in quite a few works in the context of secure multiparty computation [1], but also recently in zero-knowledge proofs [30]. For some choices of m, $\mathbb{Z}_m$ may not admit large enough exceptional sets, but this can be fixed by considering a ring extension of $\mathbb{Z}_m$ of large enough degree.
- *Homomorphic commitments.* Arguably, the biggest difficulty in extending the techniques in [3] to any ring of the form $\mathbb{Z}_m$ lies in

efficiently and securely instantiating the homomorphic commitment scheme used to hide/bind the vectors on which statements are proved. Traditionally, most commitment schemes that support any notion of homomorphism, do so modulo very structured integers. For example, constructions based on discrete-log-type assumptions typically work modulo a prime, since operations are carried out over a cyclic group. Alternatively, systems based on RSA-type assumptions tend to operate either modulo a prime, or modulo products of two primes.

To address this difficulty we present, as a contribution of potential independent interest, a novel construction of a vector commitment scheme that is homomorphic modulo m, for an *arbitrary* integer m. Our construction follows a two-step approach. First, we show how to derive a compact vector commitment scheme from any single-value commitment scheme. This consists, in a nutshell, of committing using the single-value scheme to a uniformly random linear combination of the coordinates of the desired vector, making sure to randomize the commitment with a commitment to zero. This approach is already present in other compact commitment schemes such as Pedersen's, and in this work we present an abstraction of this "compactification" technique, together with a generalization to the setting in which the modulus is any integer m.

Second, we provide an instantiation for the homomorphic single-value commitment scheme. We provide two constructions depending on the parity of m. For odd m we propose a generic template based on what we call *commitment-friendly groups*, which are essentially groups where exponentiating to all primes dividing m leads to a collision-resistant function. These groups can be used to obtain a single-value commitment scheme defined as $\textsc{Com}_{\mathsf{pk}=a}(x, r) = (a^m)^x r^m$. This is clearly hiding, and it can be proven to be binding under the assumption that p-th roots are hard to find, for any prime p dividing m. Furthermore, we instantiate commitment-friendly groups with an RSA group $\mathbb{Z}_N^*$.

The template above does not directly work for m even given that the resulting group cannot be commitment-friendly: raising to a square power clearly leads to collisions since $x^2 = (-x)^2$. To address this complication, we instead work on a subgroup of $\mathbb{Z}_N^*$, containing all elements in $\mathbb{Z}_N^*$ having Jacobi symbol 1. This way, even though it still holds that $x^2 = (-x)^2$ in this group, we can carefully choose N in such a way that this does not play any effect into the binding property.

B Special-Soundness

In this section, we recall the definition of $(k_1, \ldots, k_\mu)$-special-soundness or more precisely $(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$ special-soundness. We follow the notation of [4].

To this end, let $(\mathcal{P}, \mathcal{V})$ be a $(2\mu + 1)$-move public-coin interactive proof. Without loss of generality we may assume that the prover sends the first and final message, i.e., we assume the interactive proof has an odd number of moves (or rounds). Moreover, we assume the verifier to sample its i-th challenge from a challenge set $\mathcal{C}_i$ of cardinality N_i. The following defines a $(k_1, \ldots, k_\mu)$-tree of transcripts for $(\mathcal{P}, \mathcal{V})$ to be a set of $K = k_1 \cdots k_\mu$ protocol transcripts $(a_1, c_1, a_2, \ldots, c_\mu, a_{\mu+1})$ that are in a certain tree structure. For a graphical representation see [4].

Definition 4 (Tree of Transcripts). *Let $k_1, \ldots, k_\mu \in \mathbb{N}$. A $(k_1, \ldots, k_\mu)$-tree of transcripts for a $(2\mu+1)$-move public-coin protocol $(\mathcal{P}, \mathcal{V})$ is a set of $K = \prod_{i=1}^{\mu} k_i$ transcripts arranged in the following tree structure. The nodes in this tree correspond to the prover's messages and the edges to the verifier's challenges. Every node at depth i has precisely k_i children corresponding to k_i pairwise distinct challenges. Every transcript corresponds to exactly one path from the root node to a leaf node.*

Definition 5 ($(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$ Special-Soundness). *A $(2\mu+1)$-move public-coin protocol $(\mathcal{P}, \mathcal{V})$ for relation R, where $\mathcal{V}$ samples the i-th challenge from a set of cardinality $N_i \geq k_i$ for $1 \leq i \leq \mu$, is $(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$ special-sound if there exists a polynomial time algorithm that, on input a statement x and a $(k_1, \ldots, k_\mu)$-tree of accepting transcripts outputs a witness w such that $(x; w) \in R$. We also say $(\mathcal{P}, \mathcal{V})$ is $(k_1, \ldots, k_\mu)$-special-sound.*

It is well known that, for 3-move protocols, k-special-soundness implies knowledge soundness, but only recently it was shown that more generally, for public-coin $(2\mu + 1)$-move protocols, $(k_1, \ldots, k_\mu)$-special-soundness tightly implies knowledge soundness [4].

Theorem 7 ([4]). *A $(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$ special-sound interactive proof is knowledge sound with knowledge error*

$$\kappa = 1 - \prod_{i=1}^{\mu} \frac{N_i - k_i + 1}{N_i} \,.$$

C Some Remarks on our Commitments

Remark 5 (Using class groups). Alternatively, we can take G to be a class group. Such a group is constructed from a discriminant Δ, and it is a standard assumption that for large enough Δ, the order of the corresponding class group is hard to compute. If Δ is a prime, then the order of the group is odd, but otherwise we do not know any way to efficiently compute information on prime factors in the order. However, as we have already mentioned, if one finds a collision for ϕ_p as defined above, one can find an element of order p, and for odd p one can reasonably conjecture that this is a hard problem in class groups. The assumption on p'th roots is motivated by the fact that the group order is hard to compute, in a similar way as for RSA.

The case of $p = 2$ requires special care. The issue is that if the prime factors of Δ are known, one can compute square roots efficiently in the class group. Therefore, for even m, we need that Δ is hard to factor. One can of course use an RSA modulus as discriminant, but this provides little advantage as then it would be more efficient to do the RSA based solution directly. For an alternative, see the discussion below on trusted set-up.

Remark 6 (On trusted setup). It can be an advantage in practice if the public key of the commitment scheme can be chosen in such a way that no one knows any side information that would allow breaking the scheme. Delegating key generation to a trusted party will work, but one would clearly prefer a solution where no trusted party is needed.

For the RSA-based schemes, this cannot be completely satisfied since the factors of the modulus must be unknown to the committer, and we cannot generate a correctly formed modulus without access to the prime factors, or using a less efficient solution based on multiparty computation. However, observe that once the modulus N is generated, the rest of the public key, namely g, can be chosen "in public", since it is in fact just a random group element (either in $\mathbb{Z}_N^*$ for odd m, or in $\mathsf{J}^+(N)$ for even m). The vector commitment scheme we derived in Section 3.1 inherits this property since the n commitments in the public key are also random group elements. This can be useful, e.g., in case we have an RSA-based PKI. In such a setting we must assume to begin with that the factorization of the CA's modulus N is safe, and we can then leverage this modulus to generate the rest of the public key without trusted setup.

For class groups, one can generate the group G without trusted setup since the discriminant is public in a first place. In this case, however, it

is not possible to determine whether $G^m = G$ or not, as the order of G cannot be computed efficiently. Yet, precisely because of this, it seems reasonable to conjecture that, for odd m, elements in G^m are indistinguishable from random elements in G. Under this assumption we can choose g randomly in G and get a scheme that requires no trusted setup at all and still is computationally hiding since a random g cannot be distinguished from an m'th power.

For even m we need in addition, as mentioned above, that the discriminant is hard to factor. We can get such a scheme with no trusted setup by using a random discriminant large enough that it cannot be factored completely. This results in a scheme that is not very efficient in practice, but is still interesting from a theoretical point of view since no trusted setup is required.

Remark 7 (On q-one-way homomorphisms). In [23], the notion of q-one-way homomorphisms for a prime q is introduced. Informally, this is a homomorphism $f\colon G \mapsto H$ between two finite groups G and H such that (1) f is hard to invert and yet, (2) for any $y \in H$ it is easy to compute a preimage of y^q. A commitment is constructed based on this notion: the public key is $y \in \mathsf{Im}(f)$, and a commitment to $x \in \mathbb{Z}_q$ is of the form $y^x f(r)$, where $r \in G$ is uniformly random. It is very easy to see that this scheme satisfies our definition of a single-value commitment scheme, where $m = q$, and therefore implies a vector commitment scheme based on Theorem 1.

One example of a q-one-way homomorphism is $f(x) = g^x \bmod p$, where p is prime and $g \in \mathbb{Z}_p^*$ has order q. In this case, we recover the well-known Pedersen commitment scheme and its vector commitment variant (which in particular shows that our efficient reduction for proving binding applies to Pedersen vector commitments). Another example is $f(x) = x^q \bmod N$ for an RSA modulus N. Unfortunately, these constructions only work when q is prime, so they are not suitable for our needs, where we require a single-value commitment scheme over $\mathbb{Z}_m$, for *any* positive integer m.

D Proof of Theorem 5

Theorem 8 (Compression Mechanism, Thm. 5 re-stated). *Let n be even. Protocol Π_2 (as defined in Protocol 2) is a 3-round protocol for relation $\mathcal{X}^d$. It is perfectly complete and unconditionally knowledge sound with knowledge error $2/p^d$, where p is the smallest prime diving m. Its communication costs are*

– *From Prover to Verifier:* 2 $\mathcal{S}$*-commitments,* $n/2+2$ *elements in* $\mathcal{S}$ *and* 1 *elements in* $\mathcal{R}^d$.
– *From Verifier to Prover:* 1 *challenge in* $\mathcal{C} \subset \mathcal{S}$.

Proof. **Completeness**: Observe that $(c\mathbf{z}, \mathbf{z}) = (\mathbf{0}, \mathbf{x}_L) + c\mathbf{x} + c^2(\mathbf{x}_R, \mathbf{0})$. Completeness now follows from the homomorphic properties of $\textsc{Com}'_{pk}(\cdot)$ and the linearity of L.

3**-Special Soundness**: Let $(A, B, a, b, c_1, \mathbf{z}_1, \psi_1)$, $(A, B, a, b, c_2, \mathbf{z}_2, \psi_2)$ and $(A, B, a, b, c_3, \mathbf{z}_3, \psi_3)$ be three accepting transcripts for pairwise distinct challenges $c_1, c_2, c_3 \in \mathcal{C} \subset \mathcal{R}$. Let $(a_1, a_2, a_3) \in \mathcal{S}^3$ be such that

$$\begin{pmatrix} 1 & 1 & 1 \\ c_1 & c_2 & c_3 \\ c_1^2 & c_2^2 & c_3^2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Note that such a vector (a_1, a_2, a_3) exists because the Vandermonde matrix has determinant $(c_2 - c_1)(c_3 - c_1)(c_3 - c_2)$ and challenge differences are invertible modulo in $\mathcal{S}$.

Let $\tilde{\mathbf{z}} := \sum_{i=1}^{3} a_i(c_i\mathbf{z}_i, \mathbf{z}_i)$. Then, for some $\ell \in \mathbb{Z}$, $\textsc{Com}'_{pk}(\tilde{\mathbf{z}}, \tilde{\phi}) = P \cdot P^{\ell m}$, where $\tilde{\phi}$ can be computed by a recursive application of the randomness function R'.

Hence, by the zero-opening property, $(\tilde{\mathbf{z}}, \bar{\phi})$ is an opening of commitment $(P, y) \in \mathcal{L}_R$, where $\bar{\phi} = R'(\tilde{\mathbf{z}}, 0, \tilde{\psi}, R'(P^{-\ell}))$. By the linearity of L, it additionally follows that $L(\tilde{\mathbf{z}}) = y$, i.e., $(\tilde{\mathbf{z}}, \bar{\psi})$ is a witness for statement $(P, y) \in \mathcal{L}_{\mathcal{X}^d}$, which completes the proof. □

E Remarks on our Compressed Σ-Protocol

E.1 Non-Linear Constraints

In Section 4 we have shown how to open a homomorphism L on a committed vector, i.e., to prove that a committed vector $\mathbf{x} \in \mathbb{Z}_m^n$ or $\mathbf{x} \in \mathcal{S}^n$ satisfies the *linear* constraint $L(\mathbf{x}) = y$. In [3], it was shown how to handle non-linear constraints. Their approach is to linearize non-linearities by an arithmetic secret sharing based technique. After this linearization, the compressed Σ-protocol for opening homomorphisms can be applied in a black-box manner. However, we again require an adaptation, because our protocols are defined over a ring and not a field. In the lattice-based compressed Σ-protocols of [4] it was already shown how to handle this ring scenario. For this reason, we refer to their work for a detailed description of the linearization techniques. The resulting protocol for proving that a

committed vector satisfies an arbitrary (possibly non-linear) constraint captured by a circuit $C\colon \mathcal{S}^n \to \mathcal{S}$ is a protocol for relation

$$\{(P, y; \mathbf{x}, \gamma) : \textsc{Com}'_{pk}(\mathbf{x}, \gamma) = P,\ C(\mathbf{x}) = y\},$$

and it has a communication complexity that is logarithmic in the dimension n of the secret vector $\mathbf{x} \in \mathcal{S}^n$ and the number of multiplication gates m in the circuit C.

Remark 8. The lattice-based compressed Σ-protocols of [4] are also defined over a ring. However, they only considered rings with sufficiently large exceptional subsets, i.e., at least cardinality 3. For this reason, their protocol for proving linear statements could be defined over the base ring and the knowledge error could be made sufficiently small by a parallel repetition. In other words, the adaptations of the previous sections were not required in [4], when restricting to linear statements.

E.2 Fiat-Shamir Transformation

The compressed Σ-protocols of the previous sections are public-coin and therefore amenable to be made non-interactive via the Fiat-Shamir transformation. However, in general the security loss of the Fiat-Shamir transformation grows *exponentially* in the number of rounds. Moreover, there exist examples showing that this exponential security loss is tight. Choosing our parameters to account for the exponential security loss would negatively impact the efficiency.

Fortunately, it was recently shown that this exponential security loss does not apply to special-sound interactive proofs such as our compressed Σ-protocols [43,6]. In fact, the security loss was shown to be independent of the number of rounds.