Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics

Thomas Attema[1,2,3] and Ronald Cramer[1,2]

[1] CWI, Amsterdam, The Netherlands
[2] Leiden University, Leiden, The Netherlands
[3] TNO, The Hague, The Netherlands
thomas.attema@tno.nl, ronald.cramer@cwi.nl

Abstract. Σ-Protocols form a well-understood basis for *plug-and-play* secure algorithmics. Bulletproofs (Bünz et al., SP 2018) have been introduced as a "drop-in" for Σ-Protocols in some important applications; notably, zero-knowledge (ZK) for arithmetic circuits and range proofs, each with logarithmic communication instead of linear. At the heart of Bulletproofs is an ingenious, logarithmic-size proof of knowledge (PoK), denoted BP, showing that a compact Pedersen commitment to a long vector satisfies a quadratic equation ("an inner product relation"). However, *applications*, like those mentioned, meet with *technical difficulties*: (1) BPs are not ZK and (2) protocol theory requires "reinvention" with the quadratic constraint proved as its "pivot." This leads to practical, yet complex ZK protocols where applying natural plug-and-play intuition appears hard.

Our approach is radically different. We reconcile Bulletproofs with the theory of Σ-Protocols such that (a) applications can follow established protocol theory, thereby dispensing with the need for "reinventing" it, while (b) enjoying exactly the same communication reduction. We do this by giving a precise perspective on BPs as a *significant strengthening* of the power of Σ-protocols. We believe this novel perspective is rather useful for practical design.

Our program combines two essential components. First, we isolate a natural Σ-Protocol as *alternative pivot* that *directly* yields ZK proofs for *arbitrary linear statements*, while deploying suitable BPs to *compress* communication. We also develop some convenient utility enhancements of the pivot. Second, to enable ZK proofs of *nonlinear statements*, we integrate the pivot *as a blackbox* with a novel variation on – hitherto largely overlooked – *arithmetic secret sharing based techniques for Σ-Protocols* (ICITS 2012); this *linearizes* "all nonlinear statements" using the fact that *arbitrary* linear ones can be proved. This yields *simple* circuit ZK with logarithmic communication. Similarly for range proofs, which are now trivial. Our results are based on either of two assumptions, the Discrete Logarithm assumption, or an assumption derived from the Strong-RSA assumption.

Keywords: Σ-protocols, Bulletproofs, Zero-Knowledge, Plug-and-Play, Secure Algorithmics.

1 Introduction

The theory of Σ-Protocols provides a well-understood basis for *plug-and-play* secure algorithmics. Recently, Bulletproofs [BBB+18] have been introduced as a "drop-in replacement" for Σ-Protocols in several important applications. Notably, this includes ZK for arithmetic circuits with communication $O(\log |C| \cdot k)$ bits where $|C|$ is the circuit size[4] and k is the security parameter, down from $O(|C| \cdot k)$ bits. A similar result holds for range proofs.

At the heart of Bulletproofs is an interactive proof of knowledge between a Prover and Verifier showing that a Pedersen commitment to a vector of large length n satisfies a multi-variate polynomial equation of degree 2, defined with an inner product. We refer to this PoK by BP. Concretely, suppose $\mathbb{G}$ is a cyclic group of prime order q (denoted multiplicatively) supporting discrete-log-based cryptography. Suppose, furthermore, that $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{G}^n$ and $h \in \mathbb{G}$ (each g_i as well as h generators of $\mathbb{G}$) have been set up once-and-for-all such that, for parties that may subsequently act as provers, finding nontrivial linear relations between them is computationally as hard as computing discrete logarithms in $\mathbb{G}$. For each $\mathbf{x} \in \mathbb{Z}_q^n$, define $\mathbf{g}^{\mathbf{x}} = \prod_{i=1}^n g_i^{x_i}$. A Pedersen-commitment P to a vector $\mathbf{x} \in \mathbb{Z}_q^n$ is then computed as $P = \mathbf{g}^{\mathbf{x}} \cdot h^{\rho}$ where $\rho \in \mathbb{Z}_q$ is selected uniformly at random. This commitment is information-theoretically hiding and, on account of the set-up, computationally binding. Note that it is compact in the sense that, independently of n, a commitment is a single $\mathbb{G}$-element. Suppose that n is even and write $n = 2m$. Setting $\mathbf{x} = (\mathbf{x}_0, \mathbf{x}_1) \in \mathbb{Z}_q^m \times \mathbb{Z}_q^m$, a Bulletproof allows the prover to prove that it can open P such that the inner-product $\langle \mathbf{x}_0, \mathbf{x}_1 \rangle$ equals some value claimed by the prover.[5]

BPs stand out in that they ingeniously reduce communication to $O(\log n)$ group elements from $O(n)$ via traditional methods. Although this is at the expense of introducing logarithmic number of moves (instead of constant), its public-coin nature ensures that it can be rendered non-interactive using the Fiat-Shamir heuristic [FS86]. However, design of BP *applications* meet with a number of *technical difficulties*. First, BPs are not zero-knowledge, and second, cryptographic protocol theory has to be "reinvented" with the quadratic constraint proved as its "pivot." This leads to practical yet rather opaque, complex protocols where applying natural plug-and-play intuition appears hard.

1.1 Summary of Our Contributions and Organization of the Paper

In this work we take a radically different approach. We reconcile Bulletproofs with theory of Σ-Protocols such that (a) applications can follow (established) cryptographic protocol theory, thereby dispensing with the need for "reinventing" it, while (b) enjoying exactly the same communication reduction. We do this by giving a precise perspective on BPs as a *significant strengthening* of the

[4] Actually, the result in question only depends on the number of inputs and multiplication gates.

[5] Alternatively, this inner-product value may be taken as part of the committed vector.

power of Σ-protocols. We believe this novel perspective is rather useful for intuitive, plug-and-play design of practical secure algorithmics. Perhaps surprisingly our approach yields the same communication complexity; up to and including the constants.

We combine two essential components. First, we isolate a natural, *alternative pivot*: compact commitment with "arbitrary linear form openings". Given a Pedersen commitment to a long vector $\mathbf{x}$, consider ZKPok that the prover knows $\mathbf{x}$, while also revealing, for an *arbitrary, public, linear form* L, the scalar $L(\mathbf{x})$ correctly and nothing else. This has a simple Σ-Protocol. We then *compress* it by replacing the final (long) prover-message with an appropriate BP that the prover *knows it*. Indeed, the relation that this message is required to satisfy turns out amenable to deployment of a suitable BP. As a result, PoK and honest-verifier ZK are preserved, but *overall communication* drops from linear to logarithmic. In the process, we simplify known run-time analyses of knowledge extractors involved and give concrete estimates. On top of this, we introduce a further necessary utility enhancements. First, without harming overall complexity, we show, using the pivot as blackbox, how to open several linear form evaluations instead of just one. Second, we show how to apply these ideas in a setting where the secret, long vector is actually "dispersed" across several compact commitments, by compactifying these into a single compact commitment. This is useful in important applications. *From this point on, the only facts about the pivot that we will need is that we have access to a compact commitment scheme that allows a ZKPoK with low overall communication, showing that the prover knows the long secret committed vector and showing the correct openings of several linear evaluations on that committed vector;* the technical details do not matter anymore.

Second, the pivot's *significance* now surfaces when integrated with a novel variation on – hitherto largely overlooked – *arithmetic secret sharing based techniques for Σ-Protocols* [CDP12], inspired by MPC. These techniques allow for *linearization* of "nonlinear relations". It is here that free choice of linear forms in the pivot is fully exploited; the maps arising from our adaptation of [CDP12] do not form a well-structured subclass of maps. All in all, this yields *simple* logarithmic communication solutions for circuit ZK. Similarly for range proofs, which are now trivial to design. We also offer trade-offs, i.e., "square-root" complexity in constant rounds. Our results are based on either of two assumptions, the Discrete Logarithm assumption, or an assumption derived from the Strong-RSA assumption [BP97].

We proceed as follows. We start by outlining our program, in nearly exclusively conceptual fashion. We believe that the fact that it is possible to do so further underscores our main points. Later on we detail how this program deviates exactly from the paths taken in the recent literature.

1.2 A More Detailed View of Our Program

A. Our Pivotal Σ-Protocol
We isolate a basic Σ-protocol Π_s that, given a compact commitment to a secret

vector $\mathbf{x}$ of large length n, allows to *partially* open it. Concretely, given an *arbitrary, public, linear form* L, only the value $L(\mathbf{x})$ is released and nothing else. Briefly, the prover has a compact commitment P to a long secret vector $\mathbf{x}$. By a simple twist on basic Σ-protocol theory, the prover then selects a compact commitment A to a secret random vector $\mathbf{r}$. The prover sends, as first move, this commitment A *and* the values $y = L(\mathbf{x})$ and $y' = L(\mathbf{r})$. In the second move, the verifier sends a random challenge $c \in \mathbb{Z}_q$. In the third, final move, the prover then opens the commitment AP^c to a vector $\mathbf{z}$ (i.e., $\mathbf{z}$ is its committed vector; we leave the randomness underlying the commitment implicit here). Finally, the verifier checks the opening of the commitment and checks that $L(\mathbf{z}) = cy + y'$. The communication in this Σ-protocol is dominated by the *opening of* AP^c. The latter amounts to $O(nk)$ bits (where k is the security parameter), whereas the *remainder* of the protocol has $O(k)$ bits *in total*. That said, it is an honest-verifier zero-knowledge proof of knowledge (with unconditional soundness).

Using the pivotal Σ-protocol as a blackbox, its utility can be *enhanced*, which will be important later on. More concretely, *many linear forms* can be opened for essentially the price of a *single one*. First, by deploying a "polynomial amortization trick" (known, e.g., from MPC) we can do any number of *nullity* checks without any substantial increase in complexity. Second, building on this trick, we can extend the utility to the opening of *several* arbitrary linear forms $L_1, \ldots, L_s$ instead of a single one, at the cost of increasing the communication by exactly $s - 1$ values in $\mathbb{Z}_q$ (i.e., the evaluations of $s - 1$ additional forms). Finally, we note the entire discussion on these enhancements holds *verbatim* when we replace linear forms by *affine forms*.[6] For the details we refer to Section 5.

Note that we have identified two distinct *intractability assumptions*, each of which supports this pivot: the Discrete Logarithms assumption (as used in prior work involving Bulletproofs [BCC$^+$16, BBB$^+$18]) but also one derived from the Strong-RSA assumption (as nailed down in a recent, unpublished work [BFS19] on Bulletproofs and their improved applications). The introduction focuses on the DL assumption, but the Σ-protocol for the solution derived from the Strong-RSA assumption follows similarly. Our program can be based on either platform.

The details of our pivotal Σ-protocol can be found in Section 3, and the utility enhancements are described in Section 5.

B. Compressing the Pivot

We argue that protocol Π_s can be *compressed* using the ideas underlying Bulletproofs, yielding a protocol Π_c that has the same functionality and is still an honest-verifier zero-knowledge proof of knowledge for the relation in question, but that has communication $O(k \log n)$ bits *instead*, and $O(\log n)$ moves. Technically the compression degrades the soundness from unconditional to computational, and protocols with computational soundness are called arguments of knowledge. However, we will use the terms proof and argument of knowledge interchangeably.

[6] I.e., a linear form plus a constant.

Main compression idea. The idea is simply as follows, starting from Π_s. Suppose that P is the commitment in question. The linear forms are constants as they are part of the relation proved, so they will not be made explicit for now. Furthermore suppose that the prover has sent the message a as first move of Π_s, and that the verifier has subsequently sent challenge c as the second move. Thus, in the third –and final– move, the prover would be required to send the reply z. The verifier would, finally, apply the verification function ϕ attached to Π_s to check that $\phi(P; a, c, z) = 1$, and accept only if this is the case. To define the compressed protocol Π_c, instead of requiring the prover to send the long vector z, a suitable instantiation of Bulletproof's PoK will be deployed to let the prover convince the verifier that it *knows* some z such that $\phi(P; a, c, z) = 1$, which is much more efficient. Note that it is immaterial that the Bulletproof part is not zero knowledge as, in Π_s, the prover would have *revealed* z anyway.

This will ensure the claimed communication reduction, i.e., $O(k \log n)$ bits in $O(\log n)$ moves. We show that, as a *trade-off*, we may opt for *constant* number of rounds (instead of logarithmic) and $O(k\sqrt{n})$ communication (instead of logarithmic). But of course, in non-interactive Fiat-Shamir mode (which clearly applies here), the logarithmic variant may be preferable.

Note that this compression idea equally applies to the enhancements of the basic utility as discussed above. It gives essentially the same complexities. Of course, this assumes that the number of openings of linear forms is not too large; it is not sensitive to the number of nullity checks though.

The details of the compression idea can be found in Section 4.

Refined Analysis of Knowledge Extractors. In the theory of Σ-protocols, it is well known that *special soundness* implies knowledge soundness with knowledge error $1/q$, where q is the size of the challenge set. This result can be shown to follow from the convexity of the function $f(X) = X(X - 1/q)$ and an application of Jensen's inequality [Cra97]. Recently, and particularly for the above mentioned compressing techniques, natural generalizations of special soundness have become relevant. These more general notions of special soundness can again be shown to imply knowledge soundness. However, the proof technique using Jensen's inequality is no longer directly applicable. For this reason prior works [BCC$^+$16, BBB$^+$18] resort to heavy row type arguments without computing the exact knowledge error. Here, we show that an adaptation of the proof using Jensen's inequality does apply. This results in a simple proof and a refined analysis of the protocols in this paper.

The details of the extractor analysis can be found in Appendix A.

C. Compactifying a Vector of Commitments

Our compressed pivot may be summarized as compact commitments to long secret vectors that allow for very efficient partial openings, i.e., arbitrary linear forms applied to the secret committed vector. As we show later on this is sufficient for proving any (nonlinear) relation. To make this work, all relevant prover data (secret data vector plus secret auxiliary data, such a random coins) is required to be committed to in a *single compact commitment.*

However, in many relevant practical scenarios, we must assume that the commitment(s) to the prover's secret data vector about which something is to be proved in zero knowledge have already been produced *before* the zero knowledge protocol is run. In order to handle this, we require the prover to *compactify* these commitments together with the secret auxiliary data in a single compact commitment.

We consider two extreme scenarios: (1) the prover has a single compact commitment to the secret data vector about which some zero knowledge proof is to be conducted and (2) same, except that the prover has *individual* commitments to the coordinates of that secret data vector.

For the first scenario the prover uses new generators to commit to the auxiliary information. Using the compressed Σ-protocol, the prover shows that this is indeed a commitment that *exclusively* involves the new generators. Prover and verifier multiply the two compact commitments to obtain a single compact commitment to all relevant data.

For the second scenario, a basic (amortized) Σ-protocol shows that the prover knows openings to all individual commitments. From this basic protocol, we define a new Σ-protocol as follows. The prover appends the first message a of the basic protocol with a compact commitment containing all relevant data *and* the randomness sampled in the first move of the basic Σ-protocol. After receiving the challenge the prover's response can now be computed as a public linear form (parameterized by the challenge c) evaluated on the vector to which the prover committed. Instead of sending this message directly, the prover and verifier run the interactive protocol to open the associated linear form on the compact vector commitment. The verifier checks that the opening of the vector commitment is also an opening of the commitment in the Σ-protocol. As a result the prover has shown that it knows openings to all the individual commitments and that these openings are contained in the compact commitment together with the auxiliary data.

Note that each of these approaches offers a solution for either scenario. However the complexity of the first solution is linear in the number of commitments, whereas the complexity of the second solution is linear in the dimension of the vector. Hence, these modular utility enhancements are subject to trade-offs when designing ZK applications.

The details on the compactification of vector commitments can be found in Section 5.3.

D. Plug-and-Play Secure Algorithmics from Compressed Pivot
We will now explain the power of our compressed pivot. It will turn out that we only need *blackbox access*. Our key point is to show how to combine this with a hitherto largely overlooked part of Σ-protocol theory, namely the work of [CDP12] that shows how to prove *arbitrary constraints* on committed vectors by exploiting techniques from secure multi-party computation based on arithmetic secret sharing, more concretely, the ideas underlying the Commitment Multiplication Protocol from [CDM00]. It is this *combination* of "compact commitments with linear openings" and arithmetic secret sharing that allows for

"linearizing nonlinear relations". So this explains also why our compressed pivot does not need any "direct" provision to handle nonlinearity.

We need to make some appropriate adaptations to make this work for us here. We first outline the technique from [CDP12] and then we discuss adaptations. The work of [CDP12] considers homomorphic commitment schemes where the secret committed to is not a vector of large length, but a *single element* of $\mathbb{Z}_q$ instead. The primary result is a Σ-protocol showing the correctness of commitments to m multiplication triples $(\alpha_i, \beta_i, \gamma_i := \alpha_i\beta_i)$, with *low amortized complexity* for large m. In other words, the protocol verifies the multiplicative relations, and the costs per triple are relatively small.

Each of the α_i's (resp., the β_i's and γ_i's) is individually committed to. Their solution employs strongly-multiplicative packed-secret sharing. For instance, consider Shamir's scheme over $\mathbb{Z}_q$, with privacy parameter $t = 1$, but with secret-space dimension m. This uses random polynomials of degree $\leq m$, subject to the evaluations on the points $1, \ldots, m$ comprising the desired secret vector. Note that, for each sharing, a single random $\mathbb{Z}_q$-element is required (which can be taken as the evaluation at 0).

It is important to note that, given secret vector and random element, it holds by Lagrange Interpolation that, for each $c \in \mathbb{Z}_q$, the evaluation $f(c)$ of such polynomial $f(X)$ is some public $\mathbb{Z}_q$-linear combination over the coordinates of the secret vector and the random element. Namely, consider the map that takes $m + 1$ arbitrary evaluations on the points $0, \ldots, m$ and that outputs the unique polynomial $f(X)$ of degree $\leq m$ interpolating them to the evaluations of $f(X)$ in all other points. A transformation matrix describing this map does not correspond to a Vandermonde-matrix, but it can be determined from it.

Now, assume that $2m < q$ (for strong-multiplicativity). The protocol goes as follows.

- The prover selects a random polynomial $f(X)$ that defines a packed secret sharing of the vector $(\alpha_1, \ldots, \alpha_m)$. The prover also selects a random polynomial $g(X)$ that defines a packed secret sharing of the vector $(\beta_1, \ldots, \beta_m)$. Finally, the prover computes the product polynomial

$$h(X) := f(X)g(X)$$

  of degree $\leq 2m < q$.
- The prover commits to the random $\mathbb{Z}_q$-element for the sharing based on $f(X)$, i.e., $f(0)$, and commits to the random $\mathbb{Z}_q$-element for the sharing based on $g(X)$, i.e., $g(0)$. The prover also commits the evaluations of $h(X)$ on the points $0, \ldots, 2m$.[7]
- The prover sends these commitments to the verifier. The vectors of commitments to the multiplication triples are assumed to be part of the common input already.
- The verifier selects a random challenge $c \in \mathbb{Z}_q$ distinct from $1, \ldots, m$ and sends it to the prover.

[7] By Lagrange evaluation these points uniquely determine $h(X)$.

– By public linear combinations, both prover and verifier can compute three commitments: one to $u := f(c)$, one to $v := g(c)$ and one to $w := h(c)$. The prover opens each of these (assuming, of course, that c is in the right range). The verifier checks each of these three openings and checks whether

$$w = uv.$$

If the committed polynomials do not satisfy $f(X)g(X) = h(X)$, and under the assumption that the commitment scheme is binding, there are at most $2m$ values of c out of the $q-m$ possibilities such that the final check goes through. So a lying prover is caught with probability greater than $1 - 2m/(q - m)$. With q exponential in the security parameter and m, say, polynomial in it, this is exponentially close to 1. Honest-verifier zero-knowledge essentially follows from 1-privacy.

Our *first observation here* is as follows. *In the above protocol, the prover may as well use our compressed pivot as a blackbox.* Indeed, the entire vector $\mathbf{x}$ of data that the prover commits to in the protocol above can be committed to in a *single* compact commitment. Furthermore, all of the data *opened* to the verifier is some fixed linear form on the (long) secret committed vector $\mathbf{x}$. Indeed:

1. Each of the values u, v correspond to an opening of a public linear form applied to $\mathbf{x}$. The linear form is determined by some row in a transformation matrix as addressed above, under the convention that the form takes zeroes on the portion of the coordinates of x not relevant to the computation.
2. Similarly for the value w, except that this simply corresponds to an "evaluation of a polynomial whose coefficients are defined by a part of $\mathbf{x}$". So evaluation is a public linear form as well.

Overall, we get an honest-verifier proof of knowledge for showing correctness of m secret multiplication-triples with $O(k \log m)$ bits communication in $O(\log m)$ moves (or in constant rounds but with $O(k\sqrt{m})$ bits communication).

Our *second observation here* is as follows. Suppose we have an arithmetic circuit[8] C over $\mathbb{Z}_q$. We can easily turn the observation above into a solution for "circuit zero-knowledge". I.e., the prover convinces the verifier that the committed vector $\mathbf{x} \in \mathbb{Z}_q^n$ satisfies some constraint captured by a given circuit C which (wlog) returns 0. We note that [CDP12] also gives a solution for circuit zero-knowledge. But that one does not work for us here as it gives too large complexity. So we make some changes.

By the aforementioned compactification techniques it is now *sufficient* to consider the ZK scenario where the prover wants to demonstrate that C is satisfiable; this means that we may assume that the prover commits to all relevant data (inputs *and* all auxiliary data) in a *single* compact commitment. Other ZK scenarios, in which the prover has already commited to input data, are dealt

[8] The circuit has a single output vertex, and each addition/multiplication vertex has fan-in two, but unbounded fan-out.

with by first *compactifying* existing commitments and auxiliary information into a single compact commitment.

The prover first determines the computation graph implied by instantiating the circuit C with its input vector $\mathbf{x} \in \mathbb{Z}_q^n$. Let m be the number of multiplication gates in C, which we will handle as above. Using the compressed pivot, the prover commits to each of the coordinates of $\mathbf{x}$, to each *output* of the multiplication vertices and to the auxiliary values required to verify the multiplications. *The length γ of the committed vector equals $n + 2m + 3$.* Note that each wire, particularly each *input* of a multiplication vertex, can be accessed as affine combinations of the values committed to. The multiplication gates will be dealt with as before with polynomials $f(X), g(X)$ and $h(X)$. With this observation in hand, the protocol comes down to the opening of the affine map Φ that outputs $(C(\mathbf{x}), f(c), g(c), h(c))$ for which the verifier checks that $h(c) = f(c)g(c)$. As a result, circuit zero knowledge can be done $O(k \log \gamma)$ bits in $O(\log \gamma)$ moves. Trade-off between communication and moves applies as above.

More details on circuit ZK can be found in Section 6.

E. Range Proofs

The "polynomial trick" used in the pivotal Σ-Protocol immediately allows for a generalization to circuits $C : \mathbb{Z}_q^n \to \mathbb{Z}_q^s$ with an arbitrary number (say polynomial in k) output vertices without increasing the complexity of the compressed Σ-protocol. In fact, the overall communication complexity is independent of s. Note that the polynomial trick introduces an additional soundness error of $(s-1)/q$.

From this observation range proofs immediately follow. A prover simply considers the bit decomposition $\mathbf{x} \in \mathbb{Z}^n$ of an integer w, the length of this decomposition determines the range. Prover and verifier run the above circuit satisfiability protocol to commit to $\mathbf{x}$ and prove that $C(\mathbf{x}) = 0$ for $C : \mathbb{Z}_q^n \to \mathbb{Z}_q^n, \quad x \mapsto x * (1 - x)$, where $*$ represents the component-wise product. The nullity-check for C shows that the committed coefficients are indeed bits. The communication complexity of this range proof is $O(k \log n)$ bits. Using the techniques described in Section 5.3, this functionality can be extended to scenario where a prover has to prove that a Pedersen commitment to $v \in \mathbb{Z}_q$ is in a certain range.

The details can be found in Section 7.

F. Our Program from the Strong-RSA Assumption

Thus far we have implemented our program in the discrete log setting, starting from Pedersen commitments and their basic Σ-protocols. Besides some minor details in the compressed pivot, we show that the above discussion holds *verbatim* for a commitment scheme derived from the Strong-RSA assumption. More precisely, we show how the polynomial commitment scheme from a recent unpublished work [BFS19] can be adapted to open arbitrary linear forms. Our adaptations of the linearization techniques from [CDP12] are directly applicable to the Strong-RSA derived pivot.

The details can be found in Section 7 and Appendix D.

1.3 Comparison with Earlier Work

Traditional solutions for circuit ZK in the discrete logarithm setting have a communication complexity that is linear in the circuit size. Building on the work of Groth [Gro09], an ingenious recursive approach achieved logarithmic communication complexity [BCC+16]. At its heart lies an earlier version of the BP protocol discussed earlier. Further improvements were introduced in [BBB+18]. Recently, in an unpublished work [BFS19], Bünz, Fisch and Szepieniec show that similar results can be derived from the Strong-RSA assumption. The main merit of the Strong-RSA derived solutions is a significant reduction in the number of public parameters. In addition, the authors deploy proofs of exponentiation [Wes19] to reduce the computational complexity.

A common denominator in the aforementioned works is the use of a quadratic constraint as a main pivot. In [Gro09] a specific inner-product relation is introduced, and it is shown how basic Σ-protocols for this relation can be enhanced to achieve sub-linear communication complexity. A similar inner-production relation lies at the foundation of the logarithmic size protocols of [BCC+16], except that it also uses an earlier version of the BP idea. In [BBB+18], it is subsequently shown that a modification of the quadratic relation leads to better constants.

Furthermore, it is worth mentioning that in [BCC+16], as an intermediate stepping stone, a polynomial commitment scheme is constructed. A polynomial commitment is a commitment to the coefficient vector together with the functionality of opening the evaluation at any given point. The solution derived from the Strong-RSA assumption [BFS19] bases itself entirely on this polynomial functionality. For general relations it uses recent, but complicated, reductions [GWC19, MBKM19, XZZ+19].

Constructing protocols from quadratic constraints, either directly or via a polynomial commitment scheme, leads to a complex theory in which plug-and-play secure algorithmics appears hard. Significant effort is required to realize higher level applications such as circuit ZK or range proofs.

As for zero-knowledge, the work of [BBB+18] establishes this property at a higher level, and not, as do the other works, at the level of their main pivot, which leads to additional difficulties in designing ZK protocols.

The most significant difference between our approach and that of the aforementioned works is our simple and direct construction of a compressed pivot to open *arbitrary* linear forms and to combine this with the simple (MPC inspired) linearization techniques from [CDP12]. The compression is achieved by a suitable adaptation of the BP ideas [BBB+18], and the linearization techniques discard the need for a direct provision to handle nonlinearity. Despite the conceptual simplicity, the communication complexities of our approach are, even including the constants, equal to that of Bulletproofs [BBB+18].

Note that polynomial evaluation, as used in some of the other works, of course also comes down to the evaluation of a linear form, albeit a specific one. Therefore these approaches are not amenable to the linearization techniques we use.

2 Notation and Conventions

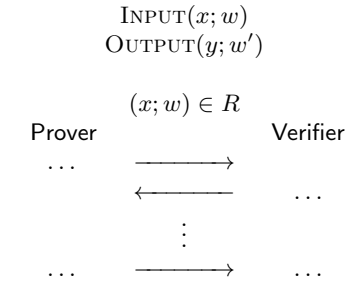
In this section, we introduce the basic notation used in the remainder of the paper. To this end let us consider dummy Protocol 1 for relation R, denoted by Π.

Let $(x; w) \in R$, then x is called a statement and w is is called a witness for x. An interactive protocol Π for relation R is a protocol that allows a prover to convince a verifier it knows a witness w for statement x. The protocol Π takes as public input x and as prover's private input w, which we write as either $\Pi(x; w)$ or, in the graphical protocol description, as $\textsc{Input}(x; w)$.

The verifier always implicitly outputs reject or accept. Optionally, the protocol can output a public string y to both verifier and prover, and a private string w' only to the prover. In this case, we write $\textsc{Output}(y; w')$.

In addition to the input and output of the protocol, the prover's claim (i.e, $(x; w) \in R$) is made explicit in the graphical protocol description.

Protocol 1 Dummy Protocol Π for Relation R



3 The Pivotal Σ-Protocol

This section formally describes the Pedersen vector commitment scheme and our pivotal Σ-protocol, as discussed in Section 1.2 (A).

Definition 1 (Pedersen Vector Commitment [Ped91]). *Let $\mathbb{G}$ be an Abelian group of prime order q, then Pedersen vector commitments are defined by the following setup and commitment phase.*

- *Setup:* $\mathbf{g} = (g_1, \ldots, g_n) \leftarrow_R \mathbb{G}^n$, $h \leftarrow_R \mathbb{G}$.
- *Commit:* $\textsc{Com} : \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathbb{G}, \quad (\mathbf{x}, \gamma) \mapsto h^\gamma \mathbf{g}^{\mathbf{x}} := h^\gamma \prod_{i=1}^n g_i^{x_i}$.

We define $\mathbf{g}^{\mathbf{x}} := \prod_{i=1}^n g_i^{x_i}$ and $\mathbf{g}^c := (g_1^c, g_2^c, \ldots, g_n^c)$ for any $\mathbf{g} \in \mathbb{G}^n$, $\mathbf{x} \in \mathbb{Z}_q^n$ and $c \in \mathbb{Z}_q$. Moreover, the component-wise product between two vectors $\mathbf{g}, \mathbf{h} \in \mathbb{G}^n$ is written as $\mathbf{g} * \mathbf{h} = (g_1 h_1, g_2 h_2, \ldots, g_n h_n)$.

Pedersen vector commitments are perfectly hiding and computationally binding under the assumption that the prover does not know a non-trivial discrete log relation between the generators $g_1, \ldots, g_n, h$.

To partially open a commitment to a linear form

$$L : \mathbb{Z}_q^n \to \mathbb{Z}_q, \quad \mathbf{x} \mapsto L(\mathbf{x}), \tag{1}$$

means that the prover wishes to reveal $L(\mathbf{x})$ together with a proof of validity without revealing any additional information on $\mathbf{x}$. Achieving this functionality amounts for the prover to send the opening $L(\mathbf{x})$ along with a ZKPoK for the relation

$$\begin{aligned} R = \left\{ \left(\mathbf{g} \in \mathbb{G}^n,\ h, P \in \mathbb{G}, L, y \in \mathbb{Z}_q; \mathbf{x} \in \mathbb{Z}_q^n, \gamma \in \mathbb{Z}_q \right) : \right. \\ \left. P = \mathbf{g}^{\mathbf{x}} h^{\gamma}, y = L(\mathbf{x}) \right\}. \end{aligned} \tag{2}$$

Protocol 2, denoted by Π_s, shows a basic Σ-protocol for relation R. Π_s was informally described in Section 1.2 (A). Theorem 1 shows that Π_s is indeed a special honest-verifier zero-knowledge (SHVZK) PoK. Both the communication costs from the prover $\mathcal{P}$ to the verifier $\mathcal{V}$ and vice versa are given. Note that in the non-interactive Fiat-Shamir [FS86] mode the communication costs from verifier to prover might be irrelevant.

Theorem 1. *Π_s is a 3-move protocol for relation R. It is perfectly complete, special honest-verifier zero-knowledge and unconditionally knowledge sound with knowledge error $1/q$. Moreover, the communication costs are*

- *$\mathcal{P} \to \mathcal{V}$: 1 element of $\mathbb{G}$ and $n + 2$ elements of $\mathbb{Z}_q$.*
- *$\mathcal{V} \to \mathcal{P}$: 1 element of $\mathbb{Z}_q$.*

4 Compressing the Pivot

This section shows how Bulletproof techniques can be applied to compress our pivotal Σ-protocol Π_s, as mentioned in Section 1.2 (B). The key observation is that sending the final message $\mathbf{z}' = (\mathbf{z}, \phi) \in \mathbb{Z}_q^{n+1}$ is actually a (trivial) proof of knowledge for the relation

$$R_1 = \left\{ (\mathbf{g}', P', L', y'; \mathbf{z}') : (\mathbf{g}')^{\mathbf{z}'} = P' \wedge y' = L'(\mathbf{z}') \right\}, \tag{3}$$

where $\mathbf{g}' = (g_1, \ldots, g_n, h) \in \mathbb{G}^{n+1}$, $P' = AP^c, y' = cy + t$ and $L'(\mathbf{z}, \phi) := L(\mathbf{z})$ for all $(\mathbf{z}, \phi)$. Hence, another PoK would also suffice, in particular a PoK with a smaller communication complexity. Moreover, it is immaterial that the PoK is zero-knowledge as the original PoK clearly is not. In [BCC+16] this observation was applied to Groth's Σ-protocol [Gro09]. The main difference is that we start with linear form relation R, whereas Groth's Σ-protocol is for a specific quadratic relation.

Protocol 2 Σ-protocol Π_s for relation R
Σ-protocol to prove correctness of a linear form evaluation.

$\textsc{Input}(\mathbf{g}, h, P, L, y; \mathbf{x}, \gamma)$

$P = \mathbf{g}^{\mathbf{x}} h^{\gamma} \in \mathbb{G}$
$y = L(\mathbf{x}) \in \mathbb{Z}_q$

Prover		Verifier
$\mathbf{r} \leftarrow_R \mathbb{Z}_q^n, \rho \leftarrow_R \mathbb{Z}_q$		
$t = L(\mathbf{r})$		
$A = \mathbf{g}^{\mathbf{r}} h^{\rho}$		
	$\xrightarrow{t, A}$	
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
$\mathbf{z} = c\mathbf{x} + \mathbf{r}$		
$\phi = c\gamma + \rho$		
	$\xrightarrow{\mathbf{z}, \phi}$	
		$\mathbf{g}^{\mathbf{z}} h^{\phi} \stackrel{?}{=} AP^c$
		$L(\mathbf{z}) \stackrel{?}{=} cy + t$

Let Π be a PoK for relation R_1. We call the new protocol obtained by replacing the final move of protocol Π_s by protocol Π the composition, and write $\Pi \diamond \Pi_s$. Since Π_s is SHVZK it immediately follows that the composition is also SHVZK.

The essence of Bulletproofs is a PoK, denoted by BP, with logarithmic communication complexity for the following inner product relation,

$$R_{\text{bullet}} = \left\{ \left(\mathbf{g}, \mathbf{h} \in \mathbb{G}^n,\ u, P \in \mathbb{G}; \mathbf{a}, \mathbf{b} \in \mathbb{Z}_p^n \right) : P = \mathbf{g}^{\mathbf{a}} \mathbf{h}^{\mathbf{b}} \wedge c = \langle \mathbf{a}, \mathbf{b} \rangle \right\}. \quad (4)$$

The quadratic relation R_{bullet} is quite similar to the relation R_1 and it turns out that minor adaptations of BP give a logarithmic size PoK for relation R_1. We will now describe the components of the BP protocol, while simultaneously adapting these to our relation R_1.

4.1 Reduction from Relation R_1 to R_2

The first step of the BP PoK is to incorporate the linear form into the Pedersen vector commitment. For this step an additional generator $k \in \mathbb{G}$ is required such that the prover does not know a discrete log relation between the generators $g_1, \ldots, g_n, h, k$. More precisely, the problem of finding a proof for relation R_1 is reduced to the problem of finding a proof for relation

$$R_2 = \left\{ \left(\mathbf{g} \in \mathbb{G}^n, k \in \mathbb{G}, P \in \mathbb{G}, L; \mathbf{x} \in \mathbb{Z}_q^n \right) : P = \mathbf{g}^{\mathbf{x}} k^{L(\mathbf{x})} \right\}. \quad (5)$$

The reduction is described in Protocol 3 and denoted by Π_1. As a stand-alone protocol, Π_1 only increases the communication complexity. However, as the authors of [BBB+18] observed, there is a more efficient PoK for relation R_2 showing the usefulness of this reduction.

Theorem 2 shows that Π_1 is an argument of knowledge for relation R_1. We include the proof of this theorem, because the authors of [BBB+18] do not consider their version of protocol Π_1 as a stand-alone protocol.

Theorem 2. *Π_1 is a 2-move protocol for relation R_1. It is perfectly complete and computationally knowledge sound with knowledge error $1/(q-1)$ under the discrete logarithm assumption. Moreover, the communication costs are:*

- *$\mathcal{P} \to \mathcal{V}$: $n+1$ elements of $\mathbb{Z}_q$.*
- *$\mathcal{V} \to \mathcal{P}$: 1 element of $\mathbb{Z}_q$.*

Proof. **Completeness** follows directly.

Knowledge soundness: We show that there exist an efficient algorithm χ that, given two accepting transcripts, either extracts a witness for R_1, or finds a non-trivial discrete log relation. So let $(c, \mathbf{x}, z)$ and $(c', \mathbf{x}', z')$ be two accepting transcripts with $c \neq c'$, then

$$\mathbf{g}^{\mathbf{x}-\mathbf{x}'} k^{z-z'} = k^{(c-c')y}. \tag{6}$$

Hence, either we have found a non-trivial discrete log relation, or $\mathbf{x} = \mathbf{x}'$ and $z - z' = (c - c')y$. In the latter case, it follows by the linearity of L that $L(\mathbf{x}) = (z - z')/(c - c') = y$. Hence, we have found a witness $\mathbf{x}$ for relation R_1. From basic Σ-protocol theory the existence of an extractor now follows, which proves the theorem.

Protocol 3 Argument of Knowledge Π_1 for R_1
Reduction from relation R_1 to relation R_2.

$\textsc{Input}(\mathbf{g}, P, L, y; \mathbf{x})$

$P = \mathbf{g}^{\mathbf{x}} \in \mathbb{G}$
$y = L(\mathbf{x}) \in \mathbb{Z}_q$

Prover		Verifier
		$c \leftarrow_R \mathbb{Z}_q^*$
	$\xleftarrow{c}$	
$z = cy$		
	$\xrightarrow{\mathbf{x}, z}$	
		$\mathbf{g}^{\mathbf{x}} k^z \stackrel{?}{=} P k^{cy}$
		$z \stackrel{?}{=} cL(\mathbf{x})$

4.2 Logarithmic Size PoK for Linear Relation R_2

Next we deploy the main technique of the Bulletproof protocol to construct an efficient PoK for relation R_2. For simplicity let us assume that n is a power of 2. If this is not the case the vector can be padded with zeros. The protocol is recursive and in each iteration the dimension of the witness is halved until its dimension equals 2. We could add one additional step to the recursion and only send the response when the dimension equals 1. This would reduce the communication costs by one field element, but it would increase the number of group elements sent by the prover by 2.

For each vector $\mathbf{g} \in \mathbb{G}^n$, we define $\mathbf{g_L} = (g_1, \ldots, g_{n/2})$ as its left halve and $\mathbf{g_R} = (g_{n/2+1}, \ldots, g_n)$ as its right halve. The same notation is used for vectors in $\mathbb{Z}_q^n$. For a linear form $L : \mathbb{Z}_q^n \to \mathbb{Z}_q$, we define

$$\begin{aligned} L_L &: \mathbb{Z}_q^{n/2} \to \mathbb{Z}_q, \quad \mathbf{x} \mapsto L(\mathbf{x}, 0), \\ L_R &: \mathbb{Z}_q^{n/2} \to \mathbb{Z}_q, \quad \mathbf{x} \mapsto L(0, \mathbf{x}), \end{aligned} \tag{7}$$

where $(\mathbf{x}, 0), (0, \mathbf{x}) \in \mathbb{Z}_q^n$ are the vectors $\mathbf{x}$ padded with $n/2$ zeros on the right and left, respectively. Recall that the component-wise product between two vectors is denoted by $*$.

Theorem 3 shows that protocol Π_2 is an argument of knowledge for relation R_2.

Theorem 3. *Π_2 is a $(2\mu + 1)$-move protocol for relation R_2. It is perfectly complete and computationally knowledge sound, under the discrete logarithm assumption, with knowledge error*

$$\kappa = \frac{\sum_{i=1}^{\mu} 6(q-1)^{\mu-i}(q-7)^{i-1}}{(q-1)^{\mu}} \leq \frac{6\mu}{q-1}, \tag{8}$$

where $\mu = \lceil \log_2(n) \rceil - 1$. Moreover, the communication costs are:

- $\mathcal{P} \to \mathcal{V}$*:* $2\lceil \log_2(n) \rceil - 2$ *elements of* $\mathbb{G}$ *and* 2 *elements of* $\mathbb{Z}_q$.
- $\mathcal{V} \to \mathcal{P}$*:* $\lceil \log_2(n) \rceil - 1$ *elements of* $\mathbb{Z}_q$.

Proof. **Completeness** follows directly.

Knowledge soundness: In the proof of Theorem 1 of [BBB+18] it was shown that there exists an efficient algorithm χ that, on input a special type of depth μ $(4, \ldots, 4)$-tree of accepting transcripts (see Appendix A.2), either computes a non-trivial discrete log-relation, or a witness for relation R_2. If χ were to succeed on any $(4, \ldots, 4)$-tree, the theorem, with a slightly different knowledge error, would immediately follow from Lemma 2 (Appendix A.2). However, χ only succeeds if at every level of the tree, every node has four distinct children $c_1, \ldots, c_4$, which satisfy the additional constraint that $c_i^2 \neq c_j^2$ for all $i \neq j$. This additional condition only requires a minor adaptation of Lemma 2 after which the theorem follows. We omit the details.

Protocol 4 Compressed Argument of knowledge Π_2 for R_2

$\textsc{Input}(\mathbf{g}, k, P, L; \mathbf{x})$

$P = \mathbf{g}^{\mathbf{x}} k^{L(\mathbf{x})}$

Prover		Verifier
$A = \mathbf{g}_R^{\mathbf{x}_L} k^{L_R(\mathbf{x}_L)}$ $B = \mathbf{g}_L^{\mathbf{x}_R} k^{L_L(\mathbf{x}_R)}$		
	$\xrightarrow{A,B}$	
		$c \leftarrow_R \mathbb{Z}_q^*$
	$\xleftarrow{c}$	
	$\mathbf{g}' := \mathbf{g_L}^{c^{-1}} * \mathbf{g_R}^{c} \in \mathbb{G}^{n/2}$ $P' := A^{c^2} P B^{c^{-2}}$ $L' := c^{-1} L_L + c L_R$	
$\mathbf{z} = c\mathbf{x_L} + c^{-1}\mathbf{x_R}$		
if $(\mathbf{z} \in \mathbb{Z}_q^2)$:	$\xrightarrow{\mathbf{z}}$	$(\mathbf{g}')^{\mathbf{z}} \mathbf{k}^{L'(\mathbf{z})} \stackrel{?}{=} P'$
else :	Run Π_2 on $(\mathbf{g}', k, P', L'; \mathbf{z})$	

4.3 Composing the Building Blocks

The compressed Σ-protocol Π_c is the composition of all previously mentioned protocols, i.e., $\Pi_c = \Pi_2 \diamond \Pi_1 \diamond \Pi_s$. Theorem 4 shows that Π_c is indeed a SHVZK argument of knowledge for relation R with a logarithmic communication complexity.

Theorem 4. *Π_c is a $(2\mu + 3)$-move protocol for relation R. It is perfectly complete, special honest-verifier zero-knowledge and computationally knowledge sound, under the discrete logarithm assumption, with knowledge error*

$$\begin{aligned} \kappa &= \frac{2(q-1)^{\mu+1} + (q-2)\sum_{i=1}^{\mu} 6(q-1)^{\mu+1-i}(q-7)^{i-1}}{q(q-1)^{\mu+1}}, \\ &\leq \frac{6\mu + 2}{q-1}, \end{aligned} \tag{9}$$

where $\mu = \lceil \log_2(n+1) \rceil - 1$. Moreover, the communication costs are:

- $\mathcal{P} \to \mathcal{V}$: $2\lceil \log_2(n+1) \rceil - 1$ *elements of* $\mathbb{G}$ *and* 3 *elements of* $\mathbb{Z}_q$.
- $\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(n+1) \rceil + 1$ *elements of* $\mathbb{Z}_q$.

Proof. **Completeness** follows directly from the completeness of Π_s, Π_1 and Π_2.

SHVZK follows directly from SHVZK of Π_s. The simulator for Π_c namely runs the simulator for Π_s and continues with honest executions of Π_1 and Π_2.

Knowledge soundness again follows from a minor adaptation of Lemma 2 (Appendix A.2).

4.4 A Remark on Sublinear Communication Complexity

For his protocols Groth [Gro09] made the observation that there is a trade-off between the communication complexity and the number of rounds. A similar trade-off applies to our situation. Protocol Π_2 achieves a logarithmic communication complexity at the cost of a logarithmic number of rounds. The protocol recursively divides the witness into two parts, left and right. This idea is easily generalized to the situation in which the witness $\mathbf{x} \in \mathbb{Z}_q^n$ is divided into k parts.

For simplicity we assume n to be a power of k. A quick inspection of this generalization shows that instead of the two group elements A and B in the first round of Π_2, the prover has to sent $2k-2$ group elements. Recursing the protocol $\log_k(n) - 1$ times results in a total communication of $(2k-2)\log_k(n) - 2k + 2$ elements of $\mathbb{G}$ and k elements of $\mathbb{Z}_q$ from prover to verifier. It is easily seen that these communication costs are minimized for $k = 2$, justifying the choices of [BCC+16, BBB+18].

In contrast, $k = \sqrt{n/2}$ results in a constant round protocol with sublinear communication costs of $\sqrt{2n} - 2$ elements of $\mathbb{G}$ and $\sqrt{2n}$ elements of $\mathbb{Z}_q$ from $\mathcal{P}$ to $\mathcal{V}$. Of course, in the non-interactive Fiat-Shamir mode the logarithmic variant might be preferable.

5 The Compressed Pivot as a Blackbox

From this point on, the only facts about the pivot that we need is that we have access to a compact vector commitment scheme that allows a prover to open *arbitrary* linear forms. Hence, we assume blackbox access to such a pivot. First, we treat the utility enhancements mentioned in Section 1.2 (A). Second, we describe the compactification techniques as discussed in Section 1.2 (C).

We use the following notation. We write $[\mathbf{x}]$ for a compact commitment to a vector $\mathbf{x} \in \mathbb{Z}_q^n$, and for a (public) linear form L we write $\Pi_{\text{OPEN}}([\mathbf{x}], L; \mathbf{x})$ for the interactive protocol that reveals $L(\mathbf{x})$ and nothing else to the verifier. Recall that our notation $\Pi_{\text{OPEN}}([\mathbf{x}], L; \mathbf{x})$ means that interactive protocol Π_{OPEN} takes as public input $[\mathbf{x}]$ and L and as prover's private input $\mathbf{x}$. The communication costs of Π_{OPEN} are equal to the cost of the underlying interactive protocol (Π_c) plus 1 field element from $\mathcal{P}$ to $\mathcal{V}$ (the output of L), unless of course the output is known in advance.

At this point, the implementation details of the compact commitment scheme do not matter anymore. However, when we give concrete knowledge errors and communication costs it is implicitly assumed that $[\cdot]$ is instantiated with Pedersen vector commitments and compressed Σ-protocol Π_c.

5.1 Amortized Nullity-Checks

A "polynomial amortization trick" (known, e.g., from MPC) allows us to do many nullity-checks on the committed vector $\mathbf{x}$ without a substantial increase in complexity. Consider linear forms $L_1, \ldots, L_s$ and *suppose the prover claims*

that $L_i(\mathbf{x}) = 0$ *for* $i = 1 \ldots, s$. The verifier then samples $\rho \in \mathbb{Z}_q$ uniformly at random and asks the prover to open the linear form $L(\mathbf{x}) = \sum_{i=1}^{s} L_i(\mathbf{x})\rho^{i-1}$. The opening of $L(\mathbf{x})$ equals the evaluation of some polynomial of degree at most $s-1$. If this polynomial is non-zero, it has at most $s-1$ zero's. Hence, $L(\mathbf{x}) = 0$ implies that $L_i(\mathbf{x}) = 0$ for all i with probability at least $1 - (s-1)/q$. When q is exponential and s is polynomial in the security parameter this probability is exponentially close to 1.

We write $\Pi_{\text{NULLITY}}([x], L_1, \ldots, L_s; , \mathbf{x})$ for this interactive protocol. The communication costs are equal to the costs of a single nullity check (i.e., $s = 1$) plus one additional $\mathbb{Z}_q$ element from $\mathcal{V}$ to $\mathcal{P}$ (the challenge ρ).

The above discussion holds *verbatim* when we replace the linear forms by affine forms $\Phi_1, \ldots, \Phi_s$, for which we also write $\Pi_{\text{NULLITY}}([x], \Phi_1, \ldots, \Phi_s; \mathbf{x})$.

5.2 Opening Affine Maps

Building on the above "polynomial trick" the functionality of the commitment scheme can be enhanced to accommodate the opening of arbitrary affine maps

$$\Phi : \mathbb{Z}_q^n \to \mathbb{Z}_q^s, \quad x \mapsto Ax + b, \tag{10}$$

at the cost of increasing the communication by exactly $s - 1$ values in $\mathbb{Z}_q$ in comparison to opening one linear form (i.e., the evaluations of $s - 1$ additional outputs). The protocol goes as follows. The prover reveals the evaluation $y = \Phi(\mathbf{x})$ followed by an amortized nullity check on the affine forms $\Phi_1(\mathbf{x}) - y_1, \ldots, \Phi_s(\mathbf{x}) - y_s$. For the interactive protocol that opens an affine map Φ we write $\Pi_{\text{OPEN}}([\mathbf{x}], \Phi; \mathbf{x})$.

5.3 Compactifying a Vector of Commitments

In Section 6 we will see that to prove nonlinear statements about a committed vector $\mathbf{x}$, additional auxiliary information $\mathsf{aux} \in \mathbb{Z}_q^t$ is required. This auxiliary information will in some sense *linearize the nonlinearities* such that, to prove the statement, only one affine map on the committed vector (x, aux) has to be opened. From this circuit satisfiability proofs immediately follow; a prover simply commits to the vector $\mathbf{x}$ together with the required auxiliary information aux and opens the associated affine map on $[(\mathbf{x}, \mathsf{aux})]$ to prove that $C(\mathbf{x}) = 0$.

However, in many scenarios one wishes to prove a statement about one or many existing commitments. In general we aim to find ZK protocols for relations of the form

$$\{([\mathbf{x}_1], \ldots, [\mathbf{x}_s], C; \mathbf{x}_1, \ldots, \mathbf{x}_s) : C(x_1, \ldots, x_s) = 0\}, \tag{11}$$

where $\mathbf{x}_i \in \mathbb{Z}_q^{n_i}$ for $1 \leq i \leq s$, and C is an arithmetic circuit. We show how to reduce these scenarios to the situation where there is a single compact commitment to all relevant data (input and auxiliary data). We consider two extreme cases:

1. The prover has a single compact commitment to the input vector $\mathbf{x}$, i.e., $s = 1$ and $n \in \mathbb{Z}_{>0}$.
2. The prover has individual commitments to the coordinates of the input vector $\mathbf{x}$, i.e., $s \in \mathbb{Z}_{>0}$ and $n_i = 1$ for all i.

From a theoretical perspective scenario (1) appears most natural. However, recent works mainly consider the second scenario. A running example for this second scenario is that of range proofs in which a prover wishes to show that many commitments to elements of $\mathbb{Z}_q$ are all in the range $[0, 2^{n-1}]$.

We treat these two scenarios separately and present different solutions. We note that both solutions are applicable to the general scenario and in particular to the two extreme cases. However, our solution for (1) has a communication complexity that is linear in the number of commitments s. In contrast, the solution for scenario (2) has a communication complexity that is linear in the dimension n. Hence, we accommodate various modes of use that are subject to specific trade-offs.

Case 1. The solution we present here makes use of the homomorphic properties of the Pedersen vector commitment scheme. It is therefore not enough to invoke $[\cdot]$ as a blackbox.

Let P be a Pedersen vector commitment to $\mathbf{x} \in \mathbb{Z}_q^n$ under generators $h, g_1, \ldots, g_n \in \mathbb{G}$. To append this commitment with auxiliary information $\mathsf{aux} \in \mathbb{Z}_q^t$ of the prover's choice, the prover and verifier append the set of generators with $g_{n+1}, \ldots, g_{n+s} \in \mathbb{G}$. The generators must be chosen such that the prover does not know a non-trivial discrete log relation. The prover then sends a Pedersen vector commitment Q, to $(0, \mathsf{aux}) \in \mathbb{Z}_q^{n+s}$, to the verifier. The product PQ is then a commitment to the vector $(\mathbf{x}, \mathsf{aux}) \in \mathbb{Z}_q^{n+t}$ under the appended set of generators.

This approach clearly allows for a prover to cheat, by simply including generators $g_1, \ldots, g_n$ into the commitment Q. To this end, the prover and verifier run the interactive nullity check on the first n coordinates of the commitment Q to $(0, \mathsf{aux})$. The interactive protocol that takes as public input a vector commitment $[\mathbf{x}]$, as prover's private input the vector aux, and outputs a commitment $[(\mathbf{x}, \mathsf{aux})]$ is written as $\Pi_{\textsc{Join}}([\mathbf{x}]; \mathbf{x}, \mathsf{aux})$.

In comparison to the circuit satisfiability example, this approach increases the communication complexity by (approximately) a factor 2. The overall protocol namely has to open affine maps on two commitments $[(0, \mathsf{aux})]$ and $[(\mathbf{x}, \mathsf{aux})]$. More involved techniques that do not incur this factor 2 loss can be constructed. First, one can amortize the costs of opening a linear form L on multiple vector commitments. Note that this means that we open the same linear form on all vector commitments. This technique does not suffice for our purposes since evaluating the above nullity check on commitment $[(\mathbf{x}, \mathsf{aux})]$ would reveal secret information about $\mathbf{x}$. To this end an additional masking term can be deployed. For details we refer to the full version of this paper. For the remainder of the paper, we will only consider the naive solution.

Case 2. Let us now consider the case where the prover has s individual Pedersen commitments P_i to $v_i \in \mathbb{Z}_q$ and wishes to prove that they satisfy some (nonlinear) relation. For simplicity, we will restrict ourselves to one Pedersen commitment P to $v \in \mathbb{Z}_q$. In Appendix B we show how this approach is generalized to an arbitrary number of Pedersen commitments for essentially the same communication costs.

The goal is to device an interactive protocol that takes as public input P, as prover's private input v and $\mathsf{aux} \in \mathbb{Z}_q^t$ and outputs a compact commitment to $[(v, \mathsf{aux})]$. In fact, our solution will output a commitment $[v, r, \mathsf{aux}]$ where $r \in \mathbb{Z}_q$ is a random element.

The approach is as follows. From the basic Σ-protocol for proving knowledge of an opening of the Pedersen commitment P, we construct a new protocol Π_P. The first message of the basic Σ-protocol is appended with a compact commitment $[\mathbf{y}] = [v, r, \mathsf{aux}]$, where r is the random element to which the prover committed in the first round of the Σ-protocol. After the final round of the Σ-protocol, the prover and verifier run the interactive nullity check on compact commitment $[\mathbf{y}]$ and affine form $L_c(\mathbf{x}) := cx_1 + x_2 - z$, where c is the verifier's challenge and z is the prover's response.

The protocol is formally described in Protocol 5. It outputs a vector commitment $[\mathbf{y}]$ and is a ZK protocol for the following relation

$$R_P = \{(P, [\mathbf{y}]; v, \gamma, \mathbf{y} = (y_1, \ldots, y_{t+2})) : P = g^v h^\gamma, v = y_1\}. \tag{12}$$

The discussion is summarized in Theorem 5.

Theorem 5. *Π_P is a $(2\mu + 5)$-move protocol for relation R_P. It is perfectly complete, special honest-verifier zero-knowledge and computationally knowledge sound, under the discrete logarithm assumption, with knowledge error*

$$\kappa \leq \frac{6\mu + 3}{q - 1}, \tag{13}$$

where $\mu = \lceil \log_2(t + 3) \rceil - 1$. Moreover, the communication costs are:

- $\mathcal{P} \to \mathcal{V}$*:* $2\lceil \log_2(t + 3) \rceil + 1$ *elements of* $\mathbb{G}$ *and* 5 *elements of* $\mathbb{Z}_q$.
- $\mathcal{V} \to \mathcal{P}$*:* $\lceil \log_2(t + 3) \rceil + 2$ *elements of* $\mathbb{Z}_q$.

Proof (Sketch). **Completeness** and **SHVZK** follow from the associated properties of the Σ-protocol and Π_{OPEN}.

Knowledge soundness: From the fact for any two distinct challenges we can define the linear form

$$L(\mathbf{x}) := \frac{L'_c - L_{c'}}{c - c'}(\mathbf{x}) = x_1, \tag{14}$$

it follows that the witness computed by the extractors of the Σ-protocol and Π_{NULLITY} satisfies the desired relation. The knowledge error follows from Lemma 3.

Note that this solution only requires blackbox access to the vector commitment scheme $[\cdot]$ and that it is immaterial that P is a Pedersen commitment. Any other commitment scheme with a Σ-protocol that satisfies some linearity constraints will suffice. In particular, the response z should be computed as the evaluation of a public linear form parameterized by the challenge c.

Protocol 5 Extended Σ-protocol Π_P for Pedersen commitments
Compactify a Pedersen commitment with auxiliary information of the prover's choice.

$\textsc{Input}(P \in \mathbb{G}; v \in \mathbb{Z}_q, \gamma \in \mathbb{Z}_q, \mathsf{aux} \in \mathbb{Z}_q^t)$
$\textsc{Output}([v, r, \mathsf{aux}]; v, r, \mathsf{aux})$

$P = g^v h^\gamma$
$L_a(\mathbf{x}) := x_1 + a x_2$

Prover		Verifier
$r \leftarrow_R \mathbb{Z}_q, \rho \leftarrow_R \mathbb{Z}_q$		
$A = g^r h^\rho$		
$\mathsf{aux} \in \mathbb{Z}_q^t$		
$\mathbf{y} = (v, r, \mathsf{aux})$		
	$\xrightarrow{A, [\mathbf{y}]}$	
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
$z = cv + r$		
$\phi = \rho + c\gamma$		
	$\xrightarrow{z, \phi}$	
		$g^z h^\phi \stackrel{?}{=} AP^c$

$\Pi_{\textsc{Nullity}}([\mathbf{y}], L_c - z; \mathbf{y})$

6 Proving Nonlinear Relations via Arithmetic Circuits

Using our commitment scheme as a blackbox, we will show how to obtain zero-knowledge arguments for arbitrary arithmetic circuits $C : \mathbb{Z}_q^n \to \mathbb{Z}_q^s$. An overview of the approach has been given in Section 1.2 (D). More precisely, we will construct a ZK protocol for the following circuit satisfiability relation:

$$R_C = \{(C; \mathbf{x}) : C(\mathbf{x}) = 0\}. \tag{15}$$

By the compactification techniques of Section 5.3 it is sufficient to consider this basic circuit satisfiability scenario. We first describe our approach for the basic scenario, and subsequently summarize the protocols that are obtained from applying the compactification techniques.

6.1 Basic Circuit Satisfiability

If C is an affine map, the protocol follows directly from the blackbox functionality. That is, the prover commits to $\mathbf{x}$ and runs $\Pi_{\text{OPEN}}([\mathbf{x}], C; \mathbf{x})$. To accommodate for nonlinear circuits, containing multiplication gates, additional techniques are required. The main idea is that the prover not only commits to $\mathbf{x}$, but also to some additional auxiliary information, such that every wire of the circuit can be accessed as an affine combination of the values committed to. The resulting commitment is therefore an implicit commitment to all wires of the circuit. What remains is to prove consistency of the nonlinear relations between the values committed to. In particular, for every multiplication gate with input wires a, b and output wire c, it must hold that $a \cdot b = c$. To this end, we use techniques from secure multi-party computation to "linearize nonlinear relations". In particular, we make appropriate adaptions to the approach of [CDP12].

Let C be a circuit with m multiplication gates and let $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{Z}_q^m$ be the vectors of left inputs, right inputs, and outputs of all the multiplications gates of C evaluated in $\mathbf{x}$. Then any wire can be accessed as an affine function on $\mathbf{x}$ and $\mathbf{c}$. The commitment $[\mathbf{x}, \mathbf{c}]$ can thus be considered as a commitment to all wires of C. However, at this point there is no guarantee that $\mathbf{a} * \mathbf{b} = \mathbf{c}$.

The key idea of [CDP12] is the use of a strongly-multiplicative packed-secret sharing scheme, e.g., Shamir's scheme over $\mathbb{Z}_q$ [Sha79] with privacy parameter $t = 1$ and secret dimension m. A secret sharing of $\mathbf{a}$ ($\mathbf{b}$) uses a random polynomial $f(X) \in \mathbb{Z}_q[X]$ ($g(X)$) of degree $\leq m$, such that the evaluations at the points $1, \ldots, m$ correspond to the vector $\mathbf{a}$ ($\mathbf{b}$). By Lagrange interpolation the polynomials are uniquely determined by its evaluations at $0, \ldots, m$, and any other evaluation can be computed as a linear combination of these evaluations.

In the circuit satisfiability protocol, the prover samples random polynomials $f(X), g(X) \in \mathbb{Z}_q[X]_{\leq m}$ that define packed secret sharings of $\mathbf{a}$ and $\mathbf{b}$, respectively. Moreover, the prover defines the product polynomial $h(X) := f(X)g(X)$. Note that the evaluations of h at $1, \ldots, m$ correspond to the vector $\mathbf{c}$. Subsequently, the prover commits to the vector $(\mathbf{x}, f(0), g(0), h(0), \ldots, h(m)) \in \mathbb{Z}_q^{n+2m+3}$. As before all wires can be accessed as affine combinations of the values committed to, but in addition all evaluations of the polynomials $f(X), g(X)$ and $h(X)$ can now be accessed in a similar manner.

The verifier then asks the prover to open $C(\mathbf{x}), f(c), g(c)$ and $h(c)$ for $c \in \mathbb{Z}_q \setminus \{1, \ldots, m\}$ chosen uniformly at random, and verifies that $h(c) = f(c)g(c)$. This verification implies that $h(X) = f(X)g(X)$, and therefore $\mathbf{a} * \mathbf{b} = \mathbf{c}$, with probability at least $1 - 2m/(q - m)$. When m is polynomial and q exponential in the security parameter k, this probability is exponentially close to 1. The entire protocol is formally described in Protocol 6 and denoted by Π_{cs}.

Π_{cs} only requires blackbox access to the commitment scheme $[\cdot]$. Theorem 3 shows that when $[\cdot]$ is instantiated with Pedersen vector commitments and compressed Σ-protocol Π_c, it is a SHVZK argument of knowledge for relation R_C. The theorem also shows that the knowledge error depends on the number of multiplication gates in the circuit. If the circuit size, and thereby m, is polyno-

mial in the security parameter and q is exponential, then the knowledge error is exponentially close to 0.

Theorem 6. *Π_{cs} is a $(2\mu+5)$-move protocol for relation R_C. It is perfectly complete, special honest-verifier zero-knowledge and computationally knowledge sound, under the discrete logarithm assumption, with knowledge error*

$$\kappa \leq \frac{6\mu + 2m + s + 5}{q - m}, \tag{16}$$

where $\mu = \lceil \log_2(n + 2m + 4) \rceil - 1$. Moreover, the communication costs are:

- *$\mathcal{P} \to \mathcal{V}$: $2\lceil \log_2(n + 2m + 4) \rceil$ elements of $\mathbb{G}$ and 6 elements of $\mathbb{Z}_q$.*
- *$\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(n + 2m + 4) \rceil + 3$ elements of $\mathbb{Z}_q$.*

Proof (Sketch). **Completeness** follows directly.

Knowledge soundness: By Lagrange interpolation there exists an efficient algorithm to reconstruct a polynomial of degree t given $t+1$ evaluations. Hence, the secret sharing round and the polynomial trick in the nullity-check introduce rounds that are $2m+1$-special sound and 4-special sound, respectively. The proof now follows from Lemma 3 (Appendix A.2).

SHVZK follows from the 1-privacy of the secret sharing scheme and the fact that Π_c is SHVZK.

6.2 Circuit ZK from Compactification

Thus far we have restricted ourselves to the circuit satisfiability scenario where the prover commits to all input and auxiliary data at once. However, there is a great variate of other scenarios, where the circuit takes as input committed values. As in Section 5.3 we consider two extreme scenarios for circuit ZK:

1. Prove that $C(\mathbf{x}) = 0$ for a vector commitment $[\mathbf{x}]$ with $\mathbf{x} \in \mathbb{Z}_q^n$,
2. Prove that $C(x_1, \ldots, x_n) = 0$ for commitments $[x_i]$ with $x_i \in \mathbb{Z}_q$ for all i.

These scenarios are dealt with by compactifying the commitments into a single compact commitment to all relevant data. The resulting protocol for scenario (1) is denoted by $\Pi_{cs}^{(1)}$ with corresponding relation $R_C^{(1)}$ and its properties are given by Theorem 7. Note that the suboptimal reduction incurs a factor 2 increase of the communication costs that can be avoided by the more involved techniques mentioned in Section 5.3.

Theorem 7. *$\Pi_{cs}^{(1)}$ is a $(2\mu+5)$-move protocol for relation $R_C^{(1)}$. It is perfectly complete, special honest-verifier zero-knowledge and computationally knowledge sound, under the discrete logarithm assumption, with knowledge error*

$$\kappa \leq \frac{12\mu + 2m + n + s + 5}{q - m}, \tag{17}$$

where $\mu = \lceil \log_2(n + 2m + 4) \rceil - 1$. Moreover, the communication costs are

Protocol 6 Circuit satisfiability argument Π_{cs} for R_C

$\textsc{Input}(C; \mathbf{x})$

$C : \mathbb{Z}_q^n \to \mathbb{Z}_q^s$
$C(\mathbf{x}) = 0$

Prover		Verifier
$f, g \leftarrow_R \mathbb{Z}_q[X]_{\leq m}$ $h(X) := f(X)g(X)$ $\mathbf{z} = (\mathbf{x}, f(0), g(0),$ $h(0), \ldots, h(2m))$		
	$\xrightarrow{[\mathbf{z}]}$	
		$c \leftarrow_R \mathbb{Z}_q \setminus \{1, \ldots, m\}$
	$\xleftarrow{c}$	
$y_1 = f(c)$ $y_2 = g(c)$ $y_3 = h(c)$		
	$\xrightarrow{y_1, y_2, y_3}$	
		$y_3 \stackrel{?}{=} y_1 y_2$

$$\Pi_{\textsc{Nullity}}\left(\begin{matrix} C(\mathbf{x}) \\ [\mathbf{z}], f(c) - y_1 \\ g(c) - y_2 \\ h(c) - y_3 \end{matrix}; \mathbf{z}\right)$$

- $\mathcal{P} \to \mathcal{V}$: $4\lceil \log_2(n + 2m + 4) \rceil - 1$ *elements of* $\mathbb{G}$ *and* 9 *elements of* $\mathbb{Z}_q$.
- $\mathcal{V} \to \mathcal{P}$: $2\lceil \log_2(n + 2m + 4) \rceil + 5$ *elements of* $\mathbb{Z}_q$.

The protocol for scenario (2) is denoted by $\Pi_{cs}^{(2)}$ with corresponding relation $R_C^{(2)}$ and its properties are given by Theorem 8. Note that in this scenario we can restrict ourselves to $n \leq 2m$. For if n is larger than the number of inputs to multiplication gates there must exist linear reductions that can be applied directly to Pedersen commitments using its homomorphic properties. Hence, the communication costs from prover to verifier can be upper-bounded by

$$2\lceil \log_2(4m + 5) \rceil + 9 \leq 2\lceil \log_2(m + 2) \rceil + 13 \tag{18}$$

elements. In comparison, Bulletproofs achieve a communication cost of $2\lceil \log(m) \rceil + 13$ elements. Hence, perhaps surprisingly, our modular plug-and-play approach almost never increases the communication costs.

Theorem 8. *$\Pi_{cs}^{(2)}$ is a $(2\mu + 5)$-move protocol for relation $R_C^{(2)}$. It is perfectly complete, special honest-verifier zero-knowledge and computationally knowledge sound, under the discrete logarithm assumption, with knowledge error*

$$\kappa \leq \frac{6\mu + n + s + 2m + 4}{q - m}, \tag{19}$$

where $\mu = \lceil \log_2(n + 2m + 5) \rceil - 1$. *Moreover, the communication costs are*

- $\mathcal{P} \to \mathcal{V}$: $2\lceil \log_2(n + 2m + 5) \rceil + 1$ *elements of* $\mathbb{G}$ *and* 8 *elements of* $\mathbb{Z}_q$.
- $\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(n + 2m + 5) \rceil + 4$ *elements of* $\mathbb{Z}_q$.

7 Range Proofs

As mentioned in Section 1.2 (E), range proofs follow as an immediate consequence of the circuit ZK protocols. We first treat the basic scenario in which the prover commits to all relevant data at one. Second, we consider the scenario where the prover wishes to convince a verifier that many Pedersen commitments are all in some range.

7.1 Basic Range Proofs

Let us consider the bit-decomposition $\mathbf{b} \in \mathbb{Z}_q^n$ of an integer in $v \in \{0, \ldots, 2^{n-1}\}$. Note that v can be computed as linear form evaluated in $\mathbf{b}$, hence a vector commitment to $\mathbf{b}$ is an implicit commitment to v.

Let $C : \mathbb{Z}_q^n \to \mathbb{Z}_q^n, \mathbf{x} \mapsto \mathbf{x} * (1 - \mathbf{x})$. Prover and verifier run Π_{cs} on input $(C; \mathbf{b})$ to obtain a ZK protocol for relation

$$R_r = \{(C; \mathbf{b}) : C(\mathbf{b}) = 0\}. \tag{20}$$

Minor improvements to a direct application of Π_{cs} can be made by observing that:

1. All multiplications gates have inputs of the form a and $1 - a$. Hence, instead of sampling a random polynomial g for the right inputs of multiplication gates we take $g = 1 - f$.
2. All outputs of multiplications gates are 0, hence $h(1) = h(2) = \cdots = h(n) = 0$ and these values do not have to be included in the compact commitment.

The full protocol, denoted by Π_r, is described in Protocol 8 (Appendix C). Theorem 9 shows that Π_r is a SHVZK argument of knowledge for relation R_r.

Theorem 9. *Π_r is a $(2\mu + 5)$-move protocol for relation R_r. It is perfectly complete, special honest-verifier zero-knowledge and computationally knowledge sound, under the discrete logarithm assumption, with knowledge error*

$$\kappa \leq \frac{6\mu + 3n + 3}{q - n}, \tag{21}$$

where $\mu = \lceil \log_2(2n + 3) \rceil - 1$. *Moreover, the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$: $2\lceil \log_2(2n + 3) \rceil$ *elements of* $\mathbb{G}$ *and* 5 *elements of* $\mathbb{Z}_q$.
- $\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(2n + 3) \rceil + 3$ *elements of* $\mathbb{Z}_q$.

7.2 Compactifying Many Pedersen Commitments

For completeness we include the properties of the protocol $\Pi_r^{(s)}$ for a prover who wishes to prove that s Pedersen commitments to $v_1, \ldots, v_s \in \mathbb{Z}_q$ are all in a range $[0, 2^{n-1}]$. For the corresponding relation we write $R_r^{(s)}$. The protocol is constructed by deploying the techniques of Section 5.3 to first obtain a single compact commitment to all relevant data. The properties of $\Pi_r^{(s)}$ are given by the following theorem.

Theorem 10. *$\Pi_r^{(s)}$ is a $(2\mu+5)$-move protocol for relation $R_r^{(s)}$. It is perfectly complete, special honest-verifier zero-knowledge and computationally knowledge sound, under the discrete logarithm assumption, with knowledge error*

$$\kappa \leq \frac{6\mu + 3ns + 2s + 3}{q - ns}, \tag{22}$$

where $\mu = \lceil \log_2(2ns + s + 4) \rceil - 1$. Moreover, the communication costs are

- *$\mathcal{P} \to \mathcal{V}$: $2\lceil \log_2(2ns + s + 4) \rceil + 1$ elements of $\mathbb{G}$ and 7 elements of $\mathbb{Z}_q$.*
- *$\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(2ns + s + 4) \rceil + 4$ elements of $\mathbb{Z}_q$.*

8 Our Program from the Strong-RSA Assumption

In this section we describe how our program can be based on assumptions derived from the Strong-RSA assumption, as mentioned in Section 1.2 (F). We treat the main differences and refer to Appendix D and [BFS19] for more details.

A disadvantage of the Pedersen vector commitment scheme is the number of generators required. In fact, to commit to an n-dimensional vector, $n+1$ generators of the group $\mathbb{G}$ are required. Moreover, the compressed Σ-protocol Π_c has a verification time that is linear in the dimension n.

Alternatively, vector commitment schemes can be constructed via integer commitment schemes [FO97, DF02]. A commitment to the vector $\mathbf{x} \in \mathbb{Z}_q^n$ is then a commitment to an integer representation $\hat{\mathbf{x}} \in \mathbb{Z}$ of $\mathbf{x}$. The integer commitment schemes of [FO97, DF02] are constructed by using groups $\mathbb{G}$ of unknown order.

This is precisely the approach followed in a recent unpublished work of Bünz, Fisch and Szepieniec [BFS19]. They construct a polynomial commitment scheme allowing a prover to commit to a polynomial $f \in \mathbb{Z}_q[X]$ of arbitrary degree, via a unique integer representation of its coefficient vector. A commitment to such a representation only requires two group elements $g, h \in \mathbb{G}$.

The authors show how to open arbitrary evaluations $f(a) \in \mathbb{Z}_q$ of a commited polynomial without revealing any additional information about f. Their polynomial evaluation protocol uses recursive techniques similar to those used in Bulletproofs. This approach results in a logarithmic communication complexity. In addition, the authors deploy Proofs of Exponentiation (PoE) [Wes19] to achieve logarithmic verification time.

The authors refer to generic constructions that can be used to obtain more general ZK protocols from polynomial commitment schemes. However, we argue

that these constructions are overly complicated and that a stronger functionality (vector commitment scheme with linear form openings) avoids many difficulties in the design of ZK protocols. Moreover, it turns out that the protocols of [BFS19] only require minor adaptations to accommodate this stronger functionality. From this, an instantiation of the blackbox functionality of Section 5 is derived, now based on the hardness assumptions related to the Strong-RSA assumption [BP97]. The techniques of Section 6 and Section 7 directly apply, and the higher level applications inherit the logarithmic communication and computation complexity of the vector commitment scheme.

9 Acknowledgements

Thomas Attema has been supported by H2020 project No 780701 (PROMETHEUS). Ronald Cramer has been supported by ERC ADG project No 74079 (ALGSTRONGCRYPTO).

References

[BBB+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, pages 315–334. IEEE Computer Society, 2018.

[BCC+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 327–357. Springer, 2016.

[BFS19] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. Transparent snarks from DARK compilers. *IACR Cryptology ePrint Archive*, 2019:1229, 2019.

[BP97] Niko Baric and Birgit Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In Walter Fumy, editor, *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*, volume 1233 of *Lecture Notes in Computer Science*, pages 480–494. Springer, 1997.

[CDM00] Ronald Cramer, Ivan Damgård, and Ueli M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334. Springer, 2000.

[CDP12] Ronald Cramer, Ivan Damgård, and Valerio Pastro. On the amortized complexity of zero knowledge protocols for multiplicative relations. In

Adam D. Smith, editor, *Information Theoretic Security - 6th International Conference, ICITS 2012, Montreal, QC, Canada, August 15-17, 2012. Proceedings*, volume 7412 of *Lecture Notes in Computer Science*, pages 62–79. Springer, 2012.

[Cra97] Ronald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD thesis, CWI and University of Amsterdam, January 1997.

[DF02] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, pages 125–142. Springer, 2002.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, volume 1294 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 1997.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[Gro09] Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 192–208. Springer, 2009.

[GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. *IACR Cryptology ePrint Archive*, 2019:953, 2019.

[MBKM19] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. Sonic: Zero-knowledge snarks from linear-size universal and updatable structured reference strings. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 2111–2128. ACM, 2019.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

[Sha79] Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

[Wes19] Benjamin Wesolowski. Efficient verifiable delay functions. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 379–407. Springer, 2019.

[XZZ+19] Tiancheng Xie, Jiaheng Zhang, Yupeng Zhang, Charalampos Papamanthou, and Dawn Song. Libra: Succinct zero-knowledge proofs with optimal prover computation. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part III*, volume 11694 of *Lecture Notes in Computer Science*, pages 733–764. Springer, 2019.

A Extractor Analysis

This appendix describes the refined extractor analysis for different generalizations of the notion *special-soundness*.

A.1 *k*-Special Soundness

In the theory of Σ-protocols, *special soundness* means that there exists an efficient algorithm to extract a witness w for statement x from a "collision", i.e., two accepting conversations (x, a, c, z) and (x, a, c', z') with $c \neq c'$. It is well known that special soundness implies knowledge soundness with knowledge error $1/q$, where q is the size of the challenge set. This result can be shown to follow from the convexity of the function $f(X) = X(X - 1/q)$ and an application of Jensen's inequality [Cra97].

To show that a special sound protocol is knowledge sound, Cramer defines the following "collision-game". This is essentially the game played by the knowledge extractor and Lemma 1 gives a bound on the success probability when playing this game. Both the game and the lemma are almost identical to the ones found in [Cra97].

Consider a 0/1-matrix with n rows and q columns. The rows will correspond to the prover's randomness and the columns to the verifier's randomness. An entry of the matrix is 1 if the prover is able to supply an accepting response for the associated first message and challenge and 0 otherwise. Let ϵ denote the number of ones in H.

The game goes as follows. Select an entry of H uniformly at random. If this entry is a 1, select another entry of the same row uniformly at random. If this entry is again a 1 the game outputs success.

To bound the success probability of the collision-game, Jensen's inequality is used. Jensen's inequality states that if X is a real random variable and f is a continuous convex function defined on the support of X, it holds that

$$f\left(\mathbb{E}[X]\right) \leq \mathbb{E}[f(X)]. \tag{23}$$

Lemma 1 (Lemma 2.1 of [Cra97]). *Let H be a 0/1-matrix with n rows and q columns, and let ϵ denote the fraction of 1-entries in H. Suppose $\epsilon > 1/q$. Then the success probability of one iteration of the "collision-game" is greater than or equal to $\epsilon(\epsilon - 1/q)$.*

Proof. Let e_i denote the number of 1-entries in the i-th row, $i = 1 \dots n.$ and let ϵ_i denote the fraction of 1-entries in the i-th row, that is $\epsilon_i = e_i/q$. Clearly, the success-probability is equal to[9]

$$\sum_{i=1}^{n} \epsilon_i \left(\frac{q\epsilon_i - 1}{q-1} \right) \leq \sum_{i=1}^{n} \epsilon_i \left(\epsilon_i - \frac{1}{q} \right). \tag{24}$$

Now observe that $\mathbb{E}[\epsilon_i] = \epsilon$, put $f(x) = x(x - 1/q)$ on the interval $[0, 1]$ and apply Jensen's inequality.

Instead of showing how knowledge soundness follows from this lemma, we immediately consider a generalization that has recently become relevant, k-special soundness. A 3-move interactive protocol is called k-special sound, if there exists an efficient algorithm that takes as input k accepting conversations $(x, a, c_1, z_1), \dots, (x, a, c_k, z_k)$ with $c_i \neq c_j$, $\forall i \neq j$, and outputs a witness w for x.

The proof technique using Jensen's inequality is no longer directly applicable, since the associated function is no longer convex. For this reason, prior works [BCC+16, BBB+18] resort to heavy row type arguments without computing the exact knowledge error. Here, we show that an adaptation of the proof using Jensen's inequality does apply. To this end let us consider the following function.

$$f : \mathbb{R} \to \mathbb{R} : \quad x \mapsto \begin{cases} \prod_{j=0}^{k-1} \frac{q}{q-j} \left(x - \frac{j}{q} \right), & \text{if } x \geq \frac{k-1}{q}, \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

Recall that $q := |\mathcal{C}|$.

It is easily seen that f is twice-differentiable and $f''(x) \geq 0$ for all $x \in \mathbb{R} \setminus \left\{ \frac{k-1}{q} \right\}$. Moreover, for $x_0 = \frac{k-1}{q}$ it holds that

$$\lim_{x \uparrow x_0} \frac{f(x) - f(x_0)}{x - x_0} = 0 \leq \frac{q}{q-k+1} \prod_{j=0}^{k-2} \frac{k-1-j}{q-j} = \lim_{x \downarrow x_0} \frac{f(x) - f(x_0)}{x - x_0}. \tag{26}$$

Hence, f is a convex function.

Theorem 11. *Let $(\mathcal{P}, \mathcal{V})$ be a k-special sound interactive protocol for relation R and let x be some statement. Let $\mathcal{P}^*$ be a prover such that $(\mathcal{P}^*, \mathcal{V})$ accepts with probability $\epsilon(x) > \frac{k-1}{q}$. Then there exists a polynomial time extractor $\mathcal{E}$ with rewindable black-box access to $\mathcal{P}^*$ that on input x outputs a witness w for x with probability at least*

$$\prod_{j=0}^{k-1} \left(\epsilon(x) - \frac{j}{q} \right) \geq \left(\epsilon(x) - \frac{k-1}{q} \right)^k, \tag{27}$$

in at most k calls to $\mathcal{P}^*$.

[9] This is minor correction of the original proof, which incorrectly states that the success probability is equal to the right hand side of this inequality.

Proof. $\mathcal{E}$ runs $(\mathcal{P}^*, \mathcal{V})$ on a random challenge $c \in \mathcal{C}$. If $\mathcal{V}$ accepts, $\mathcal{E}$ rewinds to move 2 and samples a uniform random challenge from $\mathcal{C} \setminus \{c\}$. $\mathcal{E}$ continues until it aborts or has extracted k accepting transcripts. In the latter case, k-special soundness implies the existence of an efficient algorithm to compute a witness w. So let us now determine the success probability of $\mathcal{E}$.

Let a be any first message of $(\mathcal{P}^*, \mathcal{V})$ on input $(x; w) \in R$. Let ϵ_a be the probability that $\mathcal{P}^*$ succeeds conditioned on the first message being equal to a. Then $\mathbb{E}[\epsilon_a] = \epsilon(x)$, where the expected value is taken over all possible first messages a.

Moreover, the success probability of $\mathcal{E}$, conditioned on the first message being equal to a can easily seen to be equal to $f(\epsilon_a)$, where f is defined in Equation 25.

Hence, the unconditional success probability of $\mathcal{E}$ equals

$$\mathbb{E}[f(X)] \geq f\left(\mathbb{E}[X]\right) = f(\epsilon(x)) \geq \prod_{j=0}^{k-1} \left(\epsilon(x) - \frac{j}{q}\right), \tag{28}$$

where the first inequality follows from Jensen's inequality.

A.2 Forking Lemma

A $(2\mu + 1)$-move protocol is called $(k_1, \ldots, k_\mu)$-special sound, if there exists an efficient algorithm that computes a witness from any set of $K := \prod_{i=1}^{\mu} k_i$ accepting transcripts $(x, a, c_{1,j}, z_{1,j}, \ldots, c_{\mu,j}, z_{\mu,j})$, $1 \leq j \leq K$, that they are in a $(k_1, \ldots, k_\mu)$-tree structure. The root of a $(k_1, \ldots, k_\mu)$-tree is the first message a and every node at depth i has precisely k_i distinct children $c_1, \ldots, c_{k_i} \in \mathcal{C}$. This way we obtain precisely K paths from the leaves to the root representing the accepting transcripts. Here, we show that a $(k_1, \ldots, k_\mu)$-special sound protocol is indeed knowledge sound.

The following lemma is a refinement of the forking lemma of [BCC+16]. We follow a different extractor analysis and obtain an exact knowledge error. For notational convenience Lemma 2 assumes that all challenges are sampled from $\mathbb{Z}_q$ uniformly at random. Subsequently, Lemma 3 generalizes this to the case where the verifier samples from different subsets of $\mathbb{Z}_q$ in the different rounds of the protocol. In that lemma we only give an upper bound on the soundness error.

We must note that Bulletproof are not technically $(k_1, \ldots, k_\mu)$-special sound, as the extractor imposes an additional condition on the $(k_1, \ldots, k_\mu)$-tree of accepting transcripts. For every node at depth i, it is not enough that its children $c_1, \ldots, c_{k_i} \in \mathcal{C}$ are distinct, they must also satisfy $c_i^2 \neq c_j^2$ for all $i \neq j$. It turns out that this additional constraint does not introduce any difficulties, and the proofs are easily adapted to this scenario.

Lemma 2. *Let $(\mathcal{P}, \mathcal{V})$ be a $(k_1, \ldots, k_\mu)$-special sound $(2\mu+1)$-move interactive protocol for relation R, such that the verifier samples each challenge uniformly at random from $\mathbb{Z}_q$. Let x be some statement. Let $\mathcal{P}^*$ be a prover such that $(\mathcal{P}^*, \mathcal{V})$*

accepts with probability $\epsilon(x) > \kappa$, where

$$\kappa = \frac{\sum_{i=1}^{\mu}(k_i - 1)q^{\mu-i}\prod_{j=1}^{i-1}(q - k_j + 1)}{q^{\mu}} \leq \frac{\sum_{i=1}^{\mu}(k_i - 1)}{q}. \tag{29}$$

Then there exists a polynomial time extractor $\mathcal{E}$ with rewindable black-box access to $\mathcal{P}^*$ that on input x outputs a witness w for x with probability at least $(\epsilon(x) - \kappa)^K$ in at most K calls to $\mathcal{P}^*$, where $K = \prod_{i=1}^{\mu} k_i$.

Proof. We construct a polynomial time algorithm that generates a $(k_1, \ldots, k_\mu)$-tree of accepting transcripts with probability at least $(\epsilon(x) - \kappa)^K$ in at most K calls. The lemma then follows from the definition of $(k_1, \ldots, k_\mu)$-special soundness.

For $0 \leq m \leq \mu - 1$ and $c_i \in \mathcal{C}$ let $\textsc{Tree}(x, a, c_1, \ldots, c_m)$ be the algorithm that tries to find a $(k_{m+1}, \ldots, k_\mu)$-sub-tree after the first $2m + 1$ rounds have been fixed by $a, c_1, \ldots, c_m$. More precisely, for $m = \mu$ it simply runs $\mathcal{P}^*$ on challenges $c_1, \ldots, c_\mu$ and for $m < \mu$ it runs $\textsc{Tree}(x, a, c_1, \ldots, c_m, y_\ell)$ for $1 \leq \ell \leq k_{m+1}$ and $y_\ell \in \mathcal{C}$ sampled uniformly at random such that $y_i \neq y_j$ for all $i \neq j$. We say $\textsc{Tree}$ aborts if at any stage the verifier $\mathcal{V}$ rejects and write $\textsc{Tree} = \perp$ in this case.

For notational convenience we define $\bar{\mathbf{c}}_m := (x, a, c_1, \ldots, c_m) \in \mathcal{C}^m$. For such a vector we define $\epsilon_{\bar{\mathbf{c}}_m}$ to be the probability that $\mathcal{P}^*$ succeeds conditioned on the first $2m + 1$ rounds to coincide with $\bar{\mathbf{c}}_m$. Moreover, let us define

$$\kappa_m := \frac{\sum_{i=m+1}^{\mu}(k_i - 1)q^{\mu-i}\prod_{j=m+1}^{i-1}(q - k_j + 1)}{q^{\mu-m}}. \tag{30}$$

Finally, we let $K_m = \prod_{i=m+1}^{\mu} k_i$. We will show by induction that the success probability P_m of $\textsc{Tree}(x, a, c_1, \ldots, c_m)$ is at least $\max(\epsilon_{\bar{\mathbf{c}}_m} - \kappa_m, 0)^{K_m}$ for all $0 \leq m \leq \mu$.

For $m = \mu$ the induction hypothesis immediately follows by the definition of $\epsilon_{\bar{\mathbf{c}}_m}$. So let us assume that the success probability of $\textsc{Tree}(x, a, c_1, \ldots, c_m)$ is at least $\max(\epsilon_{\bar{\mathbf{c}}_m} - \kappa_m, 0)^{K_m}$ for all $m > M$. Then,

$$
\begin{aligned}
P_M &:= P\left(\textsc{Tree}(x,a,c_1,\ldots,c_M) \neq \perp\right), \\
&= \prod_{\ell=1}^{k_{M+1}} P\left(\textsc{Tree}(x,a,c_1,\ldots,c_M,y_\ell) \neq \perp\right), \\
&\geq \prod_{\ell=1}^{k_{M+1}} \max\left(\epsilon_{\bar{\mathbf{c}}_M,y_\ell} - \kappa_{M+1}, 0\right)^{K_{M+1}}, \\
&\overset{(1)}{\geq} \prod_{\ell=1}^{k_{M+1}} \max\left(\frac{q}{q-\ell+1}\left(\epsilon_{\bar{\mathbf{c}}_M} - \frac{\ell-1}{q}\right) - \kappa_{M+1}, 0\right)^{K_{M+1}}, \\
&= \prod_{\ell=1}^{k_{M+1}} \max\left(\frac{q}{q-\ell+1}\left(\epsilon_{\bar{\mathbf{c}}_M} - \frac{\ell-1+\kappa_{M+1}(q-\ell+1)}{q}\right), 0\right)^{K_{M+1}}, \\
&\overset{(2)}{\geq} \prod_{\ell=1}^{k_{M+1}} \left(\epsilon_{\bar{\mathbf{c}}_M} - \frac{k_{M+1}-1+\kappa_{M+1}(q-k_{M+1}+1)}{q}, 0\right)^{K_{M+1}}, \\
&= \max\left(\epsilon_{\bar{\mathbf{c}}_M} - \frac{k_{M+1}-1+\kappa_{M+1}(q-k_{M+1}+1)}{q}, 0\right)^{K_M}, \\
&\overset{(3)}{=} \max\left(\epsilon_{\bar{\mathbf{c}}_M} - \kappa_M, 0\right)^{K_M}.
\end{aligned}
\tag{31}
$$

For inequality (1) we use that y_ℓ is sampled uniformly at random from $\mathcal{C} \setminus \{y_1,\ldots,y_{\ell-1}\}$, hence

$$
\begin{aligned}
\epsilon_{\bar{\mathbf{c}}_M,y_\ell} &= \frac{q}{q-\ell+1}\left(\epsilon_{\bar{\mathbf{c}}_M} - \frac{1}{q}\sum_{j=1}^{\ell-1}\epsilon_{\bar{\mathbf{c}}_M,y_j}\right), \\
&\geq \frac{q}{q-\ell+1}\left(\epsilon_{\bar{\mathbf{c}}_M} - \frac{\ell-1}{q}\right).
\end{aligned}
\tag{32}
$$

For inequality (2) we use that $\ell \leq k_{M+1}$ and that $0 \leq \kappa_m \leq 1$ for all m. For equality (3) we use that

$$
\frac{k_{M+1}-1+\kappa_{M+1}(q-k_{M+1}+1)}{q} = \kappa_M. \tag{33}
$$

Hence, by induction the hypothesis is true for all $0 \leq m \leq \mu$. In particular, we find that $P(\textsc{Tree}(x,a)) \geq \max(\epsilon_a - \kappa, 0)^K$. Now define the convex function,

$$
f : \mathbb{R} \to \mathbb{R} : \quad x \mapsto \begin{cases} (x-\kappa)^K, & \text{if } x \geq \kappa, \\ 0, & \text{otherwise.} \end{cases} \tag{34}
$$

Then the success probability of the extractor is at least

$$
\mathbb{E}[f(\epsilon_a)] \geq f\left(\mathbb{E}[\epsilon_a]\right) = f(\epsilon(x)) = \left(\epsilon(x) - \kappa\right)^K, \tag{35}
$$

where the first inequality follows by Jensen's inequality. This proves the theorem.

Lemma 3. *Let $(\mathcal{P}, \mathcal{V})$ be a $(k_1, \ldots, k_\mu)$-special sound $(2\mu+1)$-move interactive protocol for relation R, such that the verifier samples challenge c_i in move $2i$ uniformly at random from $\mathcal{C}_i \subset \mathbb{Z}_q$ for $1 \leq i \leq \mu$. Let x be some statement. Let $n_i := |\mathcal{C}_i|$ and let $\mathcal{P}^*$ be a prover such that $(\mathcal{P}^*, \mathcal{V})$ accepts with probability $\epsilon(x) > \kappa$, where*

$$\kappa \leq \sum_{i=1}^{\mu} \frac{k_i - 1}{n_i}. \tag{36}$$

Then there exists a polynomial time extractor $\mathcal{E}$ with rewindable black-box access to $\mathcal{P}^*$ that on input x outputs a witness w for x with probability at least $(\epsilon(x) - \kappa)^K$ in at most K calls to $\mathcal{P}^*$, where $K = \prod_{i=1}^{\mu} k_i$.

B Compactifying Pedersen Commitments

Protocol 7 allows a prover to convince a verifier in ZK to have knowledge of the openings of s Pedersen commitments, i.e., it is a ZKPoK for the following relation,

$$R_p^s = \{(P_1, \ldots, P_s; v_1, \gamma_1, \ldots v_s, \gamma_s) : P_j = g^{v_j} h^{\gamma_j} \quad 1 \leq j \leq s\}. \tag{37}$$

The protocol uses a standard generalization of the Σ-protocol for individual Σ-protocols. As in Protocol 5 we adapt the standard protocol to make use of our vector commitment scheme as a blackbox. The result is that the protocol outputs, in addition, a commitment to the vector $(v_1, \ldots, v_s, \mathsf{aux}) \in \mathbb{Z}_q^{s+t}$, where the prover is free to chose the auxiliary $\mathsf{aux} \in \mathbb{Z}_q$.

Theorem 12. *Π_P^s is a $(2\mu+5)$-move protocol for relation R_P^s. It is perfectly complete, special honest-verifier zero-knowledge and computationally knowledge sound, under the discrete logarithm assumption, with knowledge error*

$$\kappa \leq \frac{6\mu + s + 2}{q - 1}, \tag{38}$$

where $\mu = \lceil \log_2(s + t + 2) \rceil - 1$. Moreover, the communication costs are:

- *$\mathcal{P} \to \mathcal{V}$: $2\lceil \log_2(s + t + 2) \rceil + 1$ elements of $\mathbb{G}$ and 5 elements of $\mathbb{Z}_q$.*
- *$\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(s + t + 2) \rceil + 2$ elements of $\mathbb{Z}_q$.*

C Range Proof

Protocol 8, denoted by Π_r, is a SHVZK argument of knowledge for relation R_r (Theorem 9), where

$$R_r = \{(C; \mathbf{b}) : C(\mathbf{b}) = 0\}, \tag{39}$$

and $C : \mathbb{Z}_q^n \to \mathbb{Z}_q^n, \mathbf{x} \mapsto \mathbf{x} * (1 - \mathbf{x})$. The polynomial f in this protocol is sampled uniformly at random from all polynomials of degree n for which the evaluations at $1, \ldots, n$ correspond to $\mathbf{b}$. The protocol is very similar to the circuit satisfiability protocol of Section 6. Minor improvements are possible due to the specific nature of the circuit C.

Protocol 7 Extended Σ-protocol Π_P^s for s Pedersen commitments

$\textsc{Input}(P_1, \ldots, P_s; v_1, \gamma_1, \ldots v_s, \gamma_s, \mathsf{aux})$
$\textsc{Output}([v_1, \ldots, v_s, r, \mathsf{aux}])$

$P_j = g^{v_j} h^{\gamma_j}$
$L_a(\mathbf{x}) := x_{s+1} + \sum_{i=1}^{s} a^i x_i$

Prover		Verifier
$r \leftarrow_R \mathbb{Z}_q, \rho \leftarrow_R \mathbb{Z}_q$ $A = g^r h^\rho$ $\mathbf{y} = (v_1, \ldots, v_s, r, \mathsf{aux})$		
	$\xrightarrow{A, [\mathbf{y}]}$	
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
$z = r + \sum_{i=1}^{s} c^i v_i$ $\phi = \rho + \sum_{i=1}^{s} c^i \gamma_i$		
	$\xrightarrow{z, \phi}$	
		$g^z h^\phi \stackrel{?}{=} A \prod_{i=1}^{s} P^{c^i}$
	$\Pi_{\textsc{Nullity}}([\mathbf{y}], L_c - z; \mathbf{y})$	

Protocol 8 Range proof Π_r

$\textsc{Input}(C; \mathbf{b})$

$C : \mathbb{Z}_q^n \to \mathbb{Z}_q^n,$
$\mathbf{x} \mapsto \mathbf{x} * (1 - \mathbf{x})$
$C(\mathbf{b}) = 0$

Prover		Verifier
$f \leftarrow_R \mathbb{Z}_q[X]_{\geq n}$ $h(X) = f(X)(1 - f(X))$ $\mathbf{z} = (\mathbf{b}, f(0), h(0),$ $h(n+1) \ldots, h(2n))$	$\xrightarrow{[\mathbf{z}]}$	
		$c \leftarrow_R \mathbb{Z}_q \setminus \{1, \ldots, n\}$
	$\xleftarrow{c}$	
$y_1 = f(c)$ $y_2 = h(c)$		
	$\xrightarrow{y_1, y_2}$	
		$y_2 \stackrel{?}{=} y_1(1 - y_1)$
	$\Pi_{\textsc{Nullity}} \begin{pmatrix} C(\mathbf{b}) \\ [\mathbf{z}], f(c) - y_1; \mathbf{z} \\ h(c) - y_2 \end{pmatrix}$	

D Strong-RSA Assumption

In this appendix we informally sketch the approach of [BFS19] along with our adaptations to allows for the opening of arbitrary linear forms.

D.1 Integer Commitment Scheme

We briefly recall the integer commitment scheme of [DF02]. The commitment space of this scheme is a group $\mathbb{G}$ of unknown order, such as an RSA group or a class group. Although the exact order of $\mathbb{G}$ is unknown, we do assume to know an upper bound B on the order, i.e., $|\mathbb{G}| \leq B$.

The setup phase of the commitment scheme generates two random group elements $g, h \in \mathbb{G}$ such that they both generate the same subgroup of $\mathbb{G}$. In this case the distribution of h^γ for γ chosen uniformly at random from $[0, B \cdot 2^k)$, where k is the security parameter, will be exponentially close to the uniform distribution on $\langle g \rangle$. Hence for an arbitrary integer x, the element $[x] = g^x h^\gamma \in \mathbb{G}$ statistically hides x.

Intuitively, the binding property follows from the assumption that the prover does not know the order of $\mathbb{G}$. Formally, the binding property can be shown to follow from the root assumption [DF02, BFS19].

D.2 Vector Encoding

The vector encoding scheme of [BFS19] first lifts vectors $\mathbf{x} \in \mathbb{Z}_q^n$ to their unique representatives in $\mathbb{Z}\left(\frac{q-1}{2}\right)^n = \left\{\mathbf{x} \in \mathbb{Z}^n : \|\mathbf{x}\|_\infty \leq \frac{q-1}{2}\right\}$. Subsequently, for any $b \in \mathbb{Z}$ and $Q > 2b$ the following encoding is applied:

$$\text{Encode} : \mathbb{Z}(b)^n \to \mathbb{Z}, \quad \mathbf{x} \mapsto \sum_{i=1}^{n} x_i Q^{i-1}. \tag{40}$$

This encoding is injective since $Q > 2b$. For both $\mathbf{x} \in \mathbb{Z}_q^n$ and $\mathbf{x} \in \mathbb{Z}(b)^n$, we will write $\hat{\mathbf{x}} \in \mathbb{Z}$ for their integer encodings. A commitment $[\mathbf{x}]$ to a vector $\mathbf{x} \in \mathbb{Z}_q^n$ or in $\mathbf{x} \in \mathbb{Z}(b)^n$, is an integer commitment to $\hat{\mathbf{x}}$.

D.3 Σ-Protocol

The above thus generates a compact vector commitment scheme $[\cdot] : \mathbb{Z}_q^n \to \mathbb{G}$. For a linear form $L : \mathbb{Z}_q^n \to \mathbb{Z}_q$, this commitment scheme has a basic Σ-protocol for the relation

$$\begin{aligned} R_{\mathbb{Z}_q} = \big\{ \big(g, h, P \in \mathbb{G}, u \in \mathbb{Z}_q, Q \in \mathbb{Z}, L; \mathbf{x} \in \mathbb{Z}_q^n, \gamma \in \mathbb{Z}_q\big) : \\ P = g^{\hat{\mathbf{x}}} h^\gamma, L(\mathbf{x}) = \mathbf{u}, Q > q \big\}. \end{aligned} \tag{41}$$

The main differences between this Σ-protocol and protocol Π_s from Section 3 is that the protocol is statistically hiding and all exponents are sampled from subsets of $\mathbb{Z}$. For this reason, the verifier has to check that the final response is of bounded norm. A similar Σ-protocol is described in [BFS19].

Protocol 9 Compressed Σ-protocol for inner product relation $R_{\mathbb{Z}_q}$

$\textsc{Input}(g, h, P, Q, L; \mathbf{x}, \gamma)$

$P = g^{\hat{\mathbf{x}}} h^{\gamma} \in \mathbb{G}$
$u = L(\mathbf{x}) \in \mathbb{Z}_q^m$

Prover		Verifier
$\mathbf{r} \leftarrow_R \mathbb{Z}\left((q-1)^2 2^{k-2}\right)^n$		
$\rho \leftarrow_R \left[0, B \cdot 2^k\right)$		
$t = L(\mathbf{r}) \mod q$		
$A = g^{\hat{\mathbf{r}}} h^{\rho}$		
	$\xrightarrow{t, A}$	
		$c \leftarrow_R \left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$
	$\xleftarrow{c}$	
$\mathbf{z} = c\mathbf{x} + \mathbf{r} \in \mathbb{Z}^n$		
$\phi = c\gamma + \rho \in \mathbb{Z}$		
	$\xrightarrow{\phi, \mathbf{z}}$	
		$g^{\mathbf{z}} h^{\phi} \stackrel{?}{=} P^c A$
		$\|\mathbf{z}\|_\infty \stackrel{?}{\leq} q^2 2^{k-1}$
		$L(\mathbf{z}) \stackrel{?}{=} cu + t$

D.4 Compressed Σ-Protocol

The protocol can be compressed by observing that the response $\mathbf{z}$ is, in fact, a trivial PoK for the relation $R_{\mathbb{Z}}$.

$$\begin{aligned} R_{\mathbb{Z}} = \{(g, P \in \mathbb{G}, u \in \mathbb{Z}_q, Q, b \in \mathbb{Z}, L; \mathbf{x} \in \mathbb{Z}^n) : \\ \|\mathbf{x}\|_\infty \leq b < q, P = g^{\hat{\mathbf{x}}}, L(\mathbf{x}) = u \mod p\}. \end{aligned} \tag{42}$$

Following Bulletproof's recursive techniques a more efficient PoK for relation $R_{\mathbb{Z}}$ can be constructed. Protocol 10 shows one iteration of the recursion, repeating this recursion $O(\log n)$ times results in a logarithmic complexity. It must be noted that the bound b grows in each iteration. For this reason the encoding parameter Q has to be chosen large enough. The polynomial evaluation protocol of [BFS19] replaces the computationally expensive exponentiation after the first move ($A_R^{n/2}$) by a PoE, thereby reducing the verification time. For details we refer to [BFS19].

Another difference between this approach and the compression in the discrete log setting is that here the linear form evaluation $L(\mathbf{x})$ is not included in the commitment. For this reason the cross terms A_R and A_L have to be sent.

Protocol 10 Argument of Knowledge for relation $R_{\mathbb{Z}}$

$\textsc{Input}(g, P, u, Q, b, L; \mathbf{x})$

$$\mathbf{x} \in \mathbb{Z}(b)^n$$
$$P = g^{\hat{\mathbf{x}}}$$
$$L(\mathbf{x}) = u \mod q$$

Prover		Verifier
$A_L \leftarrow g^{\hat{\mathbf{x}}_L}$		
$A_R \leftarrow g^{\hat{\mathbf{x}}_R}$		
$u_L = L_R(\mathbf{x}_L)$		
$u_R = L_L(\mathbf{x}_R)$		
	$\xrightarrow{A_L, A_R \mathbf{u}_L, \mathbf{u}_R}$	
		$A_L A_R^{n/2} \stackrel{?}{=} P$
		$c \leftarrow_R \left[-\frac{p-1}{2}, \frac{p-1}{2}\right]$
	$\xleftarrow{c}$	
$\mathbf{z} = c\mathbf{x}_L + \mathbf{x}_R$		
	$\xrightarrow{\mathbf{z}}$	$g^{\hat{\mathbf{z}}} \stackrel{?}{=} A_L^c A_R$
		$(L_L + cL_R)(\mathbf{z}) \stackrel{?}{=}$
		$cu + c^2 u_L + u_R$
		$0 \stackrel{?}{\leq} b \stackrel{?}{<} \frac{Q}{2}$
		$\|\mathbf{z}\|_\infty \stackrel{?}{<} b\frac{q+1}{2}$