Efficient Short Adaptive NIZK for NP
Draft, September 28, 2014

Helger Lipmaa

University of Tartu, Estonia

Abstract. In Eurocrypt 2013, Gennaro et al. proposed an efficient *non-adaptive* short QAP-based NIZK argument for Circuit-SAT, where non-adaptivity means that the CRS depends on the statement to be proven. While their argument can be made adaptive by using universal circuits, this increases the prover computation by a logarithmic multiplicative factor. By following the QAP-based approach, we propose an efficient product argument, and then use it together with a modified shift argument of Fauzi et al. in the modular framework of Groth to design an *adaptive* short NIZK argument for Subset-Sum and several other NP-complete languages that has the same complexity parameters as the QAP-based non-adaptive argument, resulting in the first adaptive short NIZK arguments for NP where the prover computation is dominated by a linear number of cryptographic operations. We also construct the most efficient known range argument.

Keywords: Circuit-SAT, CRS, interpolation, NIZK, numerical NP-complete languages, product argument, quadratic arithmetic program, range proof

1 Introduction

Motivation. A common approach to design cryptographic protocols, secure in the malicious model, is to (i) design a protocol, secure in the semi-honest model, and (ii) accompany it with zero-knowledge proofs [GMR85] that the every single step of the protocol is correctly followed by the corresponding participant. More broadly, a zero-knowledge proof allows the prover to convince the verifier that some statement holds true, without revealing any side information. By using a non-interactive zero knowledge (NIZK) proof [BFM88], the prover can create a single proof π that some claim is true, without leaking any information, and then forward it to many different verifiers who can independently verify the truth of the claim without communicating with the prover. This is important in many applications like e-voting or e-auctions, where one cannot trust the voter or the bidder to be online every time the claim has to be verified.

Moreover, since the same proof can be transferred to and then verified by many independent verifiers many times, it should be as short as possible. It is well-known that sublinear-length proofs can only be computationally sound, i.e., arguments. Ideally, an argument should consist only of a few (say) group elements. For the same reason, an argument should also be efficient to verify. On the other hand, construction of the argument can be somewhat less efficient, since it is only done once. Still, prover-efficiency is important, e.g., in a situation where a single server has to create many arguments to different clients or other servers.

Due to the importance of the problem, there has been a large quantity of work on constructing efficient zero-knowledge proofs for various tasks, including NP-complete languages. In particular, there has been a large number of previous work on constructing interactive zero-knowledge proofs that can be made non-interactive in the random oracle model [FS86]. However, it is well-known that the random oracle model should only be used as a heuristic [CGH98,GK03]. The first efficient *short* NIZK argument for NP (namely, for the NP-complete language Circuit-SAT) in the standard model, i.e., in the common reference string (CRS, [BFM88]) model was proposed by Groth [Gro10].

In particular, in adaptive NIZK arguments, the CRS does not depend on the concrete instance of the language (e.g., the circuit in the case of Circuit-SAT) and thus allow re-use of the same CRS to prove satisfiability of many circuits, while in the non-adaptive case the CRS can be dependent on the instance. Since the CRS has to be generated in a secure manner and incurs additional cost, non-adaptive arguments are not sufficient for many applications.

The modular approach of Groth [Gro10], later optimized in [Lip12,FLZ13], results in adaptive short NIZK arguments for Subset-Sum with $n \log^{\omega(1)} n$ prover computation. Lipmaa [Lip14] showed how to use the results of [FLZ13] to design an adaptive Circuit-SAT argument with complexity parameters that are larger by an additional logarithmic multiplicative factor.

The most efficient known *non-adaptive* short NIZK arguments for NP-complete languages by [GGPR13] (see also [PGHR13,BSCG+13]) are based on Quadratic Arithmetic Programs. There, the prover computation is dominated by $\Theta(n)$ cryptographic operations[1], where in the case of Circuit-SAT, n is the circuit size. Other related approaches like QSP [GGPR13,Lip13] or SSP [DFGK14] have the same asymptotic complexity, and thus for simplicity we will concentrate on QAP. In *adaptive* short NIZK arguments that can obtained from non-adaptive QAP-based arguments of [GGPR13] by employing universal circuits, the prover computation is dominated by $\Theta(n \log n)$ cryptographic operations. One can ask the following natural question about bridging the gap between the non-adaptive and adaptive case:

> The Main Question of the Current Paper: *Is it possible to construct* adaptive *short NIZK arguments for NP-complete languages where the prover computation is dominated by a linear number of cryptographic operations?*

We answer this question positively in the case of Subset-Sum and several other languages, by finding a non-trivial way to combine two different approaches, the modular approach of Groth [Gro10] and the QAP-based approach of [GGPR13].

Related Work. Adaptive arguments for NP from [Gro10] and follow-up works like [Lip12,FLZ13] are built up in a modular way from a small number of basic arguments. Such *modular* arguments are based on a product argument (given commitments to vectors $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$, it holds that $c_i = a_i b_i$; a short product argument was first proposed in [Gro10], and optimized in [Lip12,FLZ13]), a permutation argument (given commitments to $\boldsymbol{a}$, $\boldsymbol{b}$, and a public permutation, it holds that $\boldsymbol{a}$ is a coordinate-wise permutation of $\boldsymbol{b}$; first proposed in [Gro10], and optimized in [Lip12,Lip14]), a shift argument (given commitments to $\boldsymbol{a}$, $\boldsymbol{b}$, it holds that $\boldsymbol{a}$ is a coordinate-wise shift of $\boldsymbol{b}$; first proposed in [FLZ13]), and possibly a small number of other arguments.

The modular approach results in adaptive NIZK arguments for NP with the communication of $\Theta(1)$ group elements. However, known modular arguments are not very efficient for the prover. E.g., the product and permutation arguments of [Lip12] have constant communication, linear verifier computation, and quadratic prover computation. More precisely, the prover has to execute a quadratic number of non-cryptographic operations and a small number of $\Theta(r_3^{-1}(n))$-wide multi-

[1] In what follows, we count communication often implicitly in group elements, and verifier computation in the number of cryptographic operations. In the case of prover computation (which is the focus of the current work), very often the number of non-cryptographic operations and cryptographic operations differs, and thus we count them separately. According to the usual but somewhat informal practice, non-cryptographic operations count cheap operations (additions or multiplications) in $\mathbb{Z}_p$, while cryptographic operations count more expensive operations (exponentiations or pairings) in a cryptographic group. The basic difference is that non-cryptographic operations are significantly (usually by more than a factor of $\log n$) more efficient than cryptographic operations

Table 1. Comparison of some of the known *adaptive* short NIZK arguments for NP-complete languages. Here, n is the number of the gates (in the case of Circuit-SAT) and the number of the integers (in the case of Subset-Sum), and $m = r_3^{-1}(n) = o(n 2^{2\sqrt{2 \log_2 n}})$. In the case of [Lip14], we included its combination with two different product arguments. Light green background denotes the best known asymptotic complexity of the concrete NP-complete language wrt. to the concrete parameter. Note that the verifier computation is almost the same in all cases.

Paper	Language	Commun.	Prover computation		Ver. comp.
			non-crypt. op.	crypt. op.	
[Gro10]	Circuit-SAT	$\Theta(1)$	$\Theta(n^2)$	$\Theta(n^2)$	$\Theta(n)$ (crypt.)
[Lip12]	Circuit-SAT	$\Theta(1)$	$\Theta(n^2)$	$\Theta(m)$	$\Theta(n)$ (crypt.)
[GGPR13]	Circuit-SAT	$\Theta(1)$	$\Theta(n \log^2 n)$	$\Theta(n \log n)$	$\Theta(n \log n)$ (non-crypt.)
[Lip14] + [Lip12]	Circuit-SAT	$\Theta(\log n)$	$\Theta(m \log^2 n)$	$\Theta(m \log n)$	$\Theta(n \log n)$ (non-crypt.)
[Lip14] + current paper	Circuit-SAT	$\Theta(\log n)$	$\Theta(n \log^2 n)$	$\Theta(n \log n)$	$\Theta(n \log n)$ (non-crypt.)
[FLZ13]	Subset-Sum	$\Theta(1)$	$\Theta(m \log n)$	$\Theta(m)$	$\Theta(n)$ (crypt.)
Current paper	Subset-Sum	$\Theta(1)$	$\Theta(n \log n)$	$\Theta(n)$	$\Theta(n)$ (crypt.)

exponentiations, where $r_3(N)$ is the size of the densest progression-free set [TV06] in $[1 \,..\, N]$. Multi-exponentiation can be significantly sped up by using algorithms from [Str64,Pip80]. The number of *non-cryptographic* operations in the prover computation in the product argument of [Lip12] can be decreased to $\Theta(r_3^{-1}(n) \log r_3^{-1}(n))$ by using the Fast Fourier Transform, see [FLZ13].

As shown in [FLZ13], one can construct adaptive short NIZK arguments for other, including several NP-complete, languages by using a small number of product and shift arguments. As shown in [Lip14], by using $\Theta(\log n)$ product and shift arguments, one can build a permutation argument, and then construct a Circuit-SAT argument as in [Gro10]. Since the shift argument of [FLZ13] is very efficient, in such arguments, all complexity parameters are strongly dominated by these of the product argument. Hence, an important open question is to further optimize the product argument. A more efficient product argument will result in more efficient adaptive arguments for different NP-complete languages but also in a more efficient range argument [CLZ12,FLZ13]. The modular framework of Groth [Gro10] of constructing complex arguments from more basic arguments is sufficiently powerful to allow construction of efficient NIZK for many other languages — given an efficient product argument.

Finding explicit progression-free sets with large r_3 (and thus small r_3^{-1}) function is one of the best known hard problems in additive combinatorics [TV06] (it is listed as the first classical open problem in [CL07]). By a recent breakthrough of Elkin [Elk11] that improved a long-standing result of Behrend [Beh46], $r_3^{-1}(n) = o(n 2^{2\sqrt{2 \log_2 n}})$. However, for any practical size of n, Elkin's construction is quite inefficient, and thus a better choice is to choose the progression-free set of Erdős and Turán [ET36] with $r_3^{-1}(n) \approx n^{\log_3 2}$. Explicit lower bounds on $r_3^{-1}(n)$ [San11,Blo14] seem to indicate that the approach of using progression-free sets has hit its limits.

Another framework for short NIZK arguments was proposed in [GGPR13]. They proposed an efficient NIZK argument for a NP-complete language QAP (Arithmetic Program), and then constructed an efficient reduction from QAP to more standard languages like (arithmetic) Circuit-SAT. The verification of a QAP instance can be written as a parallel verification of several quadratic equations between input and output bits. This can be rewritten as the verification that $(V\boldsymbol{a} + \boldsymbol{v}) \circ (W\boldsymbol{b} + \boldsymbol{w}) = Y\boldsymbol{c} + \boldsymbol{y}$, where V, W and Y are matrices, $\boldsymbol{v}$, $\boldsymbol{w}$, $\boldsymbol{y}$, $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ are vectors, and

$\circ$ denotes coordinate-wise multiplication of two vectors. Here, V, W, Y, $\boldsymbol{v}$, $\boldsymbol{w}$ and $\boldsymbol{y}$ are public but depend on the concrete circuit while $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$ are secret (i.e., related to the satisfying input).

In the QAP argument of [GGPR13], one first replaces every column vector of all three matrices with its interpolating polynomial, and then obtains a similar verification equation that involves quadratic tests of linear equations between certain polynomials. Finally, one replaces every polynomial $f(X)$ with its short garbled version $g_1^{f(\sigma)}$, where σ is a secret key and g_1 is a generator of the bilinear group. A cryptographic argument is then used to show that if a quadratic test holds between the short garbled versions, then it also must hold between the original polynomials. See [PGHR13,BSCG$^+$13,Lip13,DFGK14] for various improvements.

The QAP-based approach results in *non-adaptive* NIZK arguments that require $\Theta(n \log n)$ non-cryptographic operations (computation of polynomial interpolation and multiplication) and $\Theta(n)$ cryptographic operations (computation of $g_1^{f(\sigma)}$ from values $g_1^{\sigma^i}$ in the CRS) by the prover. These arguments are non-adaptive since in addition, the CRS has to contain elements of type $\hat{g}_1^{f(\sigma)}$ — for a different generator $\hat{g}_1$ — that depend on the matrices V, W and Y, and thus on the QAP instance (circuit in the case of Circuit-SAT) satisfiability of which the prover aims to prove.

While non-adaptive arguments are sufficient in applications like verifiable computation where the function has been fixed beforehand [GGPR13,PGHR13], they are obviously not sufficient in many other applications, since[2] the generation of every single CRS takes time $\Omega(n)$ and the storage of every single CRS takes space $\Omega(n)$. Moreover, in most circumstances, one requires a trusted third party (that can be emulated by using secure multi-party computation or secure hardware) to create the CRS. Thus, it is desirable to have a single re-usable CRS that can be used to prove the truth of many different instances, i.e., to have an adaptive argument.

The QAP-based arguments of [GGPR13] can be made adaptive by using universal circuits [Val76]. In this case, the CRS depends on the universal circuits (that can emulate all circuits of given size n on all possible inputs) and not on the concrete input circuit itself. However, since the size of universal circuits is $\Theta(n \log n)$, it means that the prover computation in resulting adaptive NIZK arguments is $\Theta(n \log^2 n)$ non-cryptographic operations and $\Theta(n \log n)$ cryptographic operations. Moreover, since Valiant's universal circuits incur a relatively large constant $c = 19$ in the $\Theta(\cdot)$ expression, a common approach [KS08,SS08] is to use universal circuits with the overhead of $\Theta(\log^2 n)$ but with a smaller constant $c = 1/2$ in the $\Theta(\cdot)$ expression. Then, the prover computation in the resulting adaptive NIZK arguments will be $\Theta(n \log^3 n)$ non-cryptographic operations and $\Theta(n \log^2 n)$ cryptographic operations. Nevertheless, up to now, the QAP-based approach of [GGPR13] and provides significantly better prover computation than the modular approach of [Gro10].

Finally, we note that non-adaptive QAP-based arguments have verifier computation that is dominated by $\Theta(1)$ cryptographic and $\Theta(\ell_u)$ non-cryptographic operations, where ℓ_u is the length of the public input of the circuit. This is not a problem in the non-adaptive setting where in the case of Circuit-SAT, there is no public input at all. However, in adaptive QAP-based arguments that are obtained by using universal circuits, $\ell_u = \Theta(n \log n)$ is equal to the length of the public input (the description of the original circuit) to the universal circuit.

Our Contributions. We answer the previously posed "main question" positively, by using the modular approach of [Gro10,FLZ13] together with the arguments constructed by using the approach of [GGPR13] (for product) and of [FLZ13] (for shift). More precisely, by using techniques related

[2] While [BCCT13,BSCTV14] propose methods to shorten the CRS, the incurred overhead in the prover computation is quite large. Moreover, one will still need a trusted third party to create the CRS.

to [GGPR13], we construct a significantly more efficient product argument than what was known before, together with a new commitment scheme (the *interpolating commitment scheme*). We then modify the shift argument of [FLZ13] to work with the interpolating commitment scheme. Finally, by using the product-and-shift framework of [FLZ13], we propose an efficient adaptive short NIZK argument for the NP-complete language SUBSET-SUM [Kar72,GJ79]. Since the new argument does not require universal circuits, it is by factor $\Theta(\log n)$ faster for the prover than the QAP-based adaptive argument for CIRCUIT-SAT. See Tbl. 1. We now describe our techniques and results.

New Product Argument. The new product argument follows the QAP-based blueprint of [GGPR13], being essentially a polynomial quadratic arithmetic program (QAP) for the circuit that computes n multiplications in parallel, with a few additional twists to achieve security in our setting.

The prover computation in the new product argument is dominated by three polynomial interpolations, one polynomial multiplication and one polynomial division, all over $\mathbb{Z}_p$ (where p is the order of the groups), and one $(n+1)$-wide multi-exponentiation. The prover computation is thus dominated by $\Theta(n \log n)$ non-cryptographic operations, assuming that p satisfies a mild criterion (the same criterion is necessary to obtain $\Theta(n \log^2 n)$ non-cryptographic operations in the arguments of [GGPR13]), and $\Theta(n)$ cryptographic operations. Such efficiency is impossible by using the progression-free set based approach due to the known upper bounds on $r_3(\cdot)$ [San11,Blo14].

Derivation of the new product argument results in a new trapdoor commitment scheme for integer vectors — the *interpolating commitment scheme* — that is a member of the family of commitment schemes known since at least [GJM02]. In fact, the interpolating commitment scheme is a very natural commitment scheme: commitment to $\boldsymbol{a} \in \mathbb{Z}_p^n$ is just a short garbled and randomized version $g_1^{L_{\boldsymbol{a}}(\sigma)} h^r$ of the Lagrange interpolating polynomial $L_{\boldsymbol{a}}(X)$ of $\boldsymbol{a}$ for a well-chosen h.

Under the assumption that the inputs to the product argument are commitments to some n-dimensional vectors (i.e., they belong to certain span; in the final arguments for NP-complete languages we guarantee this by using a knowledge assumption), we show that its soundness follows from the TSDH (Target Strong Diffie-Hellman, [PGHR13,DFGK14]) assumption. By following the terminology of [GOS12], this means that the product argument has the property of adaptive culpable soundness [GOS12], also known as *co-soundness* [GL07]. Previous work [Gro10,Lip12,FLZ13] proved culpable soundness of the product argument under a presumably weaker computational assumption (PDL, Power Discrete Logarithm [Lip12]) and an additional knowledge assumption (PKE, Power Knowledge of Exponent [Gro10]) used to ascertain that the argument itself belongs to a certain span. Due to that, the new product argument consists of only one group element, as compared to two in [Gro10,Lip12,FLZ13].

New Shift Argument. We construct a variant of the shift argument of [FLZ13], secure when combined with the interpolating commitment scheme. We prove that this argument satisfies culpable soundness under the PDL assumption and an extra knowledge (PKE) assumption. The shift argument only requires the prover to perform $\Theta(n)$ cryptographic and non-cryptographic operations.

Modular Argument for SUBSET-SUM. Finally, we describe a simple argument, motivated by that of [FLZ13], for the NP-complete language SUBSET-SUM. This argument can be seen as a short program in the scan vector parallel computation model [Ble90] that operates on committed vectors of length n. This short program consists of three commitments, one application of the shift argument, and three applications of the product argument.

Thus, in the new adaptive SUBSET-SUM adaptive argument, the prover computation is $\Theta(n \log n)$ non-cryptographic operations (dominated by a small number of polynomial interpolations, polynomial multiplications, and polynomial divisions) and $\Theta(n)$ cryptographic operations (dominated by a few ($\approx n$)-wide multi-exponentiations). Both parameters are better by a factor of $\Theta(\log n)$ when compared to the adaptive Circuit-SAT arguments in [GGPR13] and subsequent works. In fact, the prover computation is strongly dominated by $\Theta(n)$ cryptographic operations. The argument size is 11 group elements, and the verifier computation is dominated by 18 bilinear pairings and two $(n+1)$-wide multi-exponentiations. Multi-exponentiations are only needed to once commit to S, and thus they can be pre-computed. The CRS consists of $\Theta(n)$ group elements.

Thus, we answer the stated main question of the current paper. Importantly, the current work seems be the first one that combines QAP-based techniques of [GGPR13] with unrelated techniques to create an *asymptotically* more efficient NIZK argument.

Weaker Assumptions. Another contribution of the current paper is smaller reliance on knowledge assumptions, compared to previous papers on the modular approach; this also helps to improve on the efficiency. More precisely, the new Subset-Sum argument relies on standard versions of the PDL, TSDH and PKE assumptions, and on a presumably stronger instance of the PKE assumption needed to prove culpable soundness of the shift argument. In App. H, we design another — slightly less efficient — version of the shift argument that uses a presumably stronger computational assumption (PCDH, [GJM02,BBG05,Gro10,GGPR13]). However, the resulting Subset-Sum argument relies on a presumably weaker (more standard-looking) PKE assumption.

Other Applications. A major benefit of the modular approach of [Gro10] is its generality: one can use the new basic arguments to speed up prover computation in adaptive NIZK arguments for other languages. In particular, product argument and shift argument are suitable to prove in zero knowledge that, given a set $\boldsymbol{S} = (S_1, \dots, S_n)$ of integers possibly together with numerical parameters (say $\boldsymbol{v} = (v_1, \dots, v_n)$), there exists a subset J of $[1 \,..\, n]$, such that either the sum or product of elements of $S^* = \{S_i : i \in J\}$ satisfies some easily verifiable relation (e.g., is equal to or lesser than some public constant s). One can also prove a conjunction of such relations, for example by showing that the sum of the elements of S^* is smaller than W while the sum of $v(S_i)$ for $i \in J$ is larger than K. Such NP-complete languages include Subset-Sum but also Partition, Knapsack, Subset-Product, and Two-Processor Scheduling [GJ79].

For all such languages we can construct an adaptive short NIZK argument with the prover computation of $\Theta(n \log n)$ non-cryptographic (resp., $\Theta(n)$ cryptographic) operations. This can again be compared to $\Theta(n \log^2 n)$ (resp., $\Theta(n \log n)$) when using the QAP-based approach.

We then use the new product and shift arguments to speed up the short range argument of [CLZ12,FLZ13]. Since here the prover has a committed input, the new range argument is slightly more complex than the Subset-Sum argument; this also makes simulation of the argument slightly more complicated. However, differently from the Subset-Sum argument, the verifier computation is dominated only by $\Theta(1)$ cryptographic operations. For the sake of completeness, we provide a full description of the new range argument in App. I. Interestingly, this argument is computationally more efficient than any of the existing *short* range arguments in the random oracle model. E.g., the prover computation in the range argument of [Lip03] is dominated by the Rabin-Shallit algorithm [RS86] that takes quadratic prover computation under the Extended Riemann Hypothesis.

Fauzi et al. [FLZ13] constructed an efficient vector scan argument (given commitments to vectors $\boldsymbol{a}$, $\boldsymbol{b}$, it holds that $b_i = \sum_{j \geq i} a_j$). The vector scan parallel computation model [Ble90] (that assumes the existence of vector scan, Hadamard sum, Hadamard product, and possibly some other parallel operations) is very powerful, and can be used to implement many problems efficiently. In particular, [Lip14] proposed a permutation argument based on $\Theta(\log n)$ product and shift arguments. This can be used to construct an adaptive CIRCUIT-SAT argument that is very different from QAP-based arguments [GGPR13]. It has the same asymptotic prover computation as [GGPR13] but without using universal circuits, providing an interesting alternative. See Tbl. 1. We leave finding more of such cryptographically relevant languages for a future work.

On Input Size and $\boldsymbol{n}$. In the case of considered NP-complete languages, n is smaller than the input length. More precisely, n is the size of the integer set and the input length — in our arguments — is $N = \Theta(\kappa n)$, where $\kappa = n^{1/O(1)}$ is the security parameter. This means that in the new SUBSET-SUM argument the prover has to execute $N^{1-1/O(1)}$ cryptographic operations. Similarly, in CIRCUIT-SAT arguments, n is the circuit size, while the input size is $N = \Theta(n \log n)$. This means that in the QAP-based CIRCUIT-SAT argument, the prover computation is $\Theta(N)$ cryptographic operations.

2 Preliminaries: Security Assumptions

Recall that on input 1^κ, where κ is the security parameter, a (prime-order) *bilinear map generator* [SOK00,Jou00,BF01] returns $\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, \hat{e})$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are three multiplicative cyclic groups of prime order p (with $\log p = \Omega(\kappa)$), g_1 is a generator of $\mathbb{G}_1$, and $\hat{e}$ is an efficient bilinear map $\hat{e}\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that satisfies in particular the following two properties, where g_2 is an arbitrary generator of $\mathbb{G}_2$: (i) $\hat{e}(g_1, g_2) \neq 1$, and (ii) $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$. Thus, if $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1^c, g_2^d)$ then $ab \equiv cd \pmod{p}$. Within this paper we give genbp another input, n, and allow p (and thus also the groups) to depend on n. (The reason we handle g_1 and g_2 differently will become clear later.)

The security of the commitment scheme and of the new arguments depends on the following q-type assumptions, variants of which have been studied and used in many previous papers. The assumptions are parameterized but non-interactive in the sense that q is related to the parameters of the language (most generally, to the input length) and not to the number of the adversarial queries.

We will first give informal descriptions of the underlying assumptions, followed by formal descriptions. In all cases, an efficient adversary $\mathcal{A}$ has access to a benignly generated common reference string that contains gk together with elements of the form $g_i^{P_j(\sigma, \gamma_1, \dots, \gamma_m)}$, where g_i is a generator of $\mathbb{G}_i$, P_j are public polynomials, and σ and γ_i are random secret elements of $\mathbb{Z}_p$. Given such an input, the underlying assumptions state the following: **PDL:** it is infeasible for $\mathcal{A}$ to return σ, **TSDH:** for a given $i \in [1 \mathinner{..} m]$ and a given small subset S of $\mathbb{Z}_p$, it is infeasible for $\mathcal{A}$ to return $r \in S$ and $\hat{e}(g_1, g_2^{\gamma_i})^{1/(\sigma - r)}$, **PKE:** for a given $i \in [1 \mathinner{..} m]$, if $\mathcal{A}$ returns $(h, \hat{h}) = (g_1, g_2^{\gamma_i})^b$ for some b, then she must know a polynomial $a(X) = \sum a_i X^i$ in the span of $\{P_j(X)\}$, such that $b = a(\sigma)$.

More precisely, we assume that the adversary has access to elements of $\mathbb{G}_1$ and $\mathbb{G}_2$ that are of shape $g_1^{f(\sigma)}$ or $g_2^{f(\sigma, \boldsymbol{\gamma})}$, where g_i is a generator of $\mathbb{G}_i$, and σ and $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_m)$ are secret. The values γ_i are knowledge elements [Dam91] and are handled differently from σ. We assume that the formal variable X corresponds to the secret σ, and Y_i corresponds to γ_i. However, to simplify notation we will write (say) $f \in \mathbb{Z}_p[X]$ instead of $f \in \mathbb{Z}_p[X, \boldsymbol{Y}]$ in the cases where $f(X, \boldsymbol{Y})$ does not depend on $\boldsymbol{Y}$. We require that the adversary has only access to elements of $\mathbb{G}_1$ that are independent of $\boldsymbol{\gamma}$; however, assumptions have to take the presence of $\boldsymbol{\gamma}$ into account. In particular, some of the following assumptions take an index $i \in [1 \mathinner{..} m]$ as one of the arguments; in such cases γ_i is handled differently from the rest of $\boldsymbol{\gamma}$. We assume that the value of m (basically, the number of knowledge elements) is clear from the context; in the current paper, $m \in [0 \mathinner{..} 2]$. In the Subset-Sum argument in Sect. 7, $m = 2$, but every polynomial depends only on at most one Y_i. All known (to us) adaptive short NIZK arguments are based on q-type assumptions about genbp.

For a set of polynomials $\mathcal{F}$ that have the same domain, denote $g_i^{\mathcal{F}(\boldsymbol{a})} := (g_i^{f(\boldsymbol{a})})_{f \in \mathcal{F}}$. Let $\mathcal{F}_1$ (resp., $\mathcal{F}_2$) be a set of linearly independent low-degree univariate (resp., $(m+1)$-variate) polynomials. Then, genbp is $(\mathcal{F}_1, \mathcal{F}_2)$*-PDL (Power Discrete Logarithm, [Lip12]) secure* if for any $n \in poly(\kappa)$ and any non-uniform probabilistic polynomial-time adversary $\mathcal{A}$, the following probability is negligible in κ:

$$\Pr\left[\begin{array}{l}\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, \hat{e}) \leftarrow \mathsf{genbp}(1^\kappa, n), g_2 \leftarrow_r \mathbb{G}_2^*, \sigma \leftarrow_r \mathbb{Z}_p, \boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_m) \leftarrow_r \mathbb{Z}_p^m, \\ \mathsf{crs} \leftarrow (\mathsf{gk}; g_1^{\mathcal{F}_1(\sigma)}, g_2^{\mathcal{F}_2(\sigma, \boldsymbol{\gamma})}) : \mathcal{A}(\mathsf{crs}) = \sigma\end{array}\right] .$$

The following assumption is a variant of an earlier assumption from [BB04] and [PGHR13]. Let $\mathcal{F}_1$ (resp., $\mathcal{F}_2$) be a set of linearly independent low-degree univariate (resp., $(m+1)$-variate) polynomials. Let $i \in [1 \mathinner{..} m]$. Let $(\omega_1, \dots, \omega_n) \in \mathbb{Z}_p^n$ be distinct elements. Then, genbp is $(\mathcal{F}_1, \mathcal{F}_2, \{\omega_i\}_{i=1}^n, i)$*-TSDH (Target Strong Diffie-Hellman, [PGHR13]) secure* if for any $n \in poly(\kappa)$ and any non-uniform probabilistic polynomial-time adversary $\mathcal{A}$, the following probability is negligible in κ:

$$\Pr\left[\begin{array}{l}\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, \hat{e}) \leftarrow \mathsf{genbp}(1^\kappa, n), g_2 \leftarrow_r \mathbb{G}_2^*, \sigma \leftarrow_r \mathbb{Z}_p \setminus \{\omega_i\}_{i=1}^n, \boldsymbol{\gamma} \leftarrow_r \mathbb{Z}_p^m, \\ \mathsf{crs} \leftarrow \left(\mathsf{gk}; g_1^{\mathcal{F}_1(\sigma)}, g_2^{\mathcal{F}_2(\sigma, \boldsymbol{\gamma})}\right) : \mathcal{A}(\mathsf{crs}) = \left(j \in [1 \mathinner{..} n], \hat{e}(g_1, g_2^{\gamma_i})^{1/(\sigma - \omega_j)}\right)\end{array}\right] .$$

In [BB04], the adversary must output $(r, g_1^{1/(\sigma - r)})$ for some $r \in \mathbb{Z}_p$, given access to $g_1^{\sigma^i}$ for (say) $i \in [0 \mathinner{..} n]$. A variant where the adversary has to output a target group element was introduced in [PGHR13]. We restrict the power of the adversary by requiring r to belong to the set $\{\omega_i\}$. The $(\mathcal{F}_1, \mathcal{F}_2, \{\omega_i\}_{i=1}^n, i)$-TSDH assumption is clearly at least as strong as the $(\mathcal{F}_1, \mathcal{F}_2)$-PDL assumption for the same $(\mathcal{F}_1, \mathcal{F}_2)$. Both PDL and TSDH are variants of the uber-assumption from [BBG05].

For algorithms $\mathcal{A}$ and $X_\mathcal{A}$, we write $(y; y_X) \leftarrow (\mathcal{A} || X_\mathcal{A})(\sigma)$ if $\mathcal{A}$ on input σ outputs y, and $X_\mathcal{A}$ on the same input (including the random tape of $\mathcal{A}$) outputs y_X. Let $\mathcal{F}$ be a set of linearly independent low-degree univariate polynomials, and $\mathcal{G}_1$ (resp., $\mathcal{G}_2$) be a set of linearly independent low-degree univariate (resp., m-variate) polynomials. Let $i \in [1 \mathinner{..} m]$. Then, genbp is $(\mathcal{F}, \mathcal{G}_1, \mathcal{G}_2, i)$*-PKE (Power Knowledge of Exponent, [Gro10]) secure* if for any non-uniform probabilistic polynomial-time adversary $\mathcal{A}$ there exists a non-uniform probabilistic polynomial-time extractor $X_\mathcal{A}$, such that the following probability is negligible in κ:

$$\Pr\left[\begin{array}{l}\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, \hat{e}) \leftarrow \mathsf{genbp}(1^\kappa, n), g_2 \leftarrow_r \mathbb{G}_2^*, \sigma \leftarrow_r \mathbb{Z}_p, \\ \boldsymbol{\gamma_{-i}} = (\gamma_1, \dots, \gamma_{i-1}, \gamma_{i+1}, \dots, \gamma_m) \leftarrow_r \mathbb{Z}_p^{m-1}, \mathsf{aux} \leftarrow_r \left(g_1^{\mathcal{G}_1(\sigma)}, g_2^{\mathcal{G}_2(\sigma, \boldsymbol{\gamma}_{-i})}\right), \gamma_i \leftarrow_r \mathbb{Z}_p, \\ \mathsf{crs} \leftarrow \left(\mathsf{gk}; (g_1, g_2^{\gamma_i})^{\mathcal{F}(\sigma)}, \mathsf{aux}\right), (h, \hat{h}; (a_i)_{i=0}^n) \leftarrow (\mathcal{A} || X_\mathcal{A})(\mathsf{crs}), a(X) \leftarrow \displaystyle\sum_{i=0}^n a_i P_i(X) : \\ \hat{e}(h, g_2^{\gamma_i}) = \hat{e}(g_1, \hat{h}) \wedge h \neq g_1^{\sum_{i=0}^n a(\sigma)}\end{array}\right] .$$

Here, $\hat{e}(h, g_2^{\gamma_i}) = \hat{e}(g_1, \hat{h})$ guarantees that $(h, \hat{h}) = (g_1, g_2^{\gamma_i})^b$ for some b. It is not necessary that $\mathcal{F}$ (resp., $\mathcal{G}_i$) is linearly independent since one can remove linearly dependent polynomials from $\mathcal{F}$ (resp., $\mathcal{G}_i$) without changing the assumption. Moreover, aux can be seen as the common auxiliary input to $\mathcal{A}$ and $X_{\mathcal{A}}$ that is generated independently of γ_i; its generation it can be seen as benign auxiliary input generation [BCPR14]. A version of the PKE assumption was first defined in [Gro10]. We use an asymmetric version from [DFGK14] but for a more general sets of polynomials.

By generalizing [BB08,Gro10,Lip12], one can show that the TSDH, PDL and PKE assumptions hold in the generic bilinear group model. We emphasize that as we will see in Sect. 7, in the Subset-Sum argument of Sect. 7 most of the interesting sets $\mathcal{F}$ (or $\mathcal{F}_j$) are equal to either $\{X^i\}_{i\in[0\,..\,d]}$ or $\{Y_jX^i\}_{i\in[0\,..\,d]}$ for some integer d and $j \in [1\,..\,m]$.

3 New Trapdoor Commitment Scheme

In this section, we construct a new trapdoor commitment scheme. We first give a general construction and prove its security, and then give an instantiation of the parameters that we need in the current paper. Precise reasoning behind the recommended parameters will become clear in Sect. 5.

A trapdoor commitment scheme is a randomized cryptographic primitive in the CRS model [BFM88] that inputs a message and outputs a commitment. It consists of two efficient algorithms gencom (that outputs a CRS and a trapdoor) and com (that, given a CRS, a message and a randomizer, outputs a commitment), and must satisfy the following three security properties. **Computational binding:** without access to the trapdoor, it is intractable to open a commitment to two different messages. **Perfect hiding:** commitments of any two messages have the same distribution. **Trapdoor:** given access to the original message, the randomizer and the trapdoor, one can open the commitment to any other message. See, e.g., [Gro10] for formal definitions. We define the following pairing-based *polynomial commitment scheme.*

Definition 1 ($\{P_i\}$-Commitment Scheme). *Let $n = \mathrm{poly}(\kappa)$, $n > 0$, be an integer. Let $P_i(X) \in \mathbb{Z}_p[X]$, for $i \in [0\,..\,n]$, be $n+1$ linearly independent low-degree polynomials. First, $\mathsf{gencom}(1^\kappa, n)$ invokes the bilinear group generator to generate $\mathsf{gk} \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, \hat{e})$, picks $g_2 \leftarrow_r \mathbb{G}_2^*$, and then outputs the CRS (also known as the commitment key)*

$$\mathsf{ck} \leftarrow (\mathsf{gk}; (g_1^{P_i(\sigma)}, g_2^{\gamma_1 P_i(\sigma)})_{i=0}^n) \tag{1}$$

for $\sigma \leftarrow_r \mathbb{Z}_p \setminus \{j : P_0(j) = 0\}$ and $\gamma_1 \leftarrow_r \mathbb{Z}_p^*$. The trapdoor is equal to (σ, γ_1).

The commitment of $\boldsymbol{a} \in \mathbb{Z}_p^n$, given a randomizer $r \leftarrow_r \mathbb{Z}_p$, is $\mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r) := (g_1^{P_0(\sigma)}, g_2^{\gamma_1 P_0(\sigma)})^r \cdot \prod_{i=1}^n (g_1^{P_i(\sigma)}, g_2^{\gamma_1 P_i(\sigma)})^{a_i} \in \mathbb{G}_1 \times \mathbb{G}_2$. The validity of a commitment $(A_1, A_2^{\gamma_1})$ can be checked by verifying that $\hat{e}(A_1, g_2^{\gamma_1 P_0(\sigma)}) = \hat{e}(g_1^{P_0(\sigma)}, A_2^{\gamma_1})$. To open a commitment, the committer sends $(\boldsymbol{a}, r)$ to the verifier.

The condition $P_0(\sigma) \neq 0$ will be needed in Thm. 1 to get perfect hiding, and the condition $\gamma_1 \neq 0$ will be needed in Thm. 7 to get perfect zero knowledge.

Clearly,

$$\log_{g_1} A_1 = \log_{g_2^{\gamma_1}} A_2^{\gamma_1} = rP_0(\sigma) + \sum_{i=1}^n a_i P_i(\sigma)\ . \tag{2}$$

The second element, $A_2^{\gamma_1}$, of the commitment is known as the knowledge component [Dam91].

For a set of polynomials $\mathcal{F} \subseteq \mathbb{Z}_p[X, Y_1, \dots, Y_{m-1}]$, we define $Y_m\mathcal{F} = \{Y_m \cdot f(X, Y_1, \dots, Y_{m-1}) : f \in \mathcal{F}\} \subseteq \mathbb{Z}_p[X, Y_1, \dots, Y_m]$. Let $\mathcal{F}_{\mathsf{com},1} = \{P_i(X)\}_{i=0}^n$ and $\mathcal{F}_{\mathsf{com},2} = Y_1\mathcal{F}_{\mathsf{com},1} = \{Y_1 P_i(X)\}_{i=0}^n$.

Theorem 1. *The $\{P_i\}$-commitment scheme is perfectly hiding. If* genbp *is* $(\mathcal{F}_{\mathsf{com},1}, \mathcal{F}_{\mathsf{com},2})$*-PDL secure, then it is computationally binding. If* genbp *is* $(\mathcal{F}_{\mathsf{com},1}, \emptyset, \emptyset, 1)$*-PKE secure, then there exists an extractor that extracts the message* $\boldsymbol{a}$ *and the randomizer* r*, given* ck*, the commitment* $(A_1, A_2^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r)$ *and access to the committer's random tape.*

Proof. Perfect Hiding: since $P_0(X)$ is a non-zero polynomial (this follows from linear independence), then due to the choice of σ, $rP_0(\sigma)$ (and thus also $\log_{g_1} A_1$) is uniformly random in $\mathbb{Z}_p$. Therefore, $(A_1, A_2^{\gamma_1})$ is a uniformly random element of the multiplicative subgroup of $\mathbb{G}_1^* \times \mathbb{G}_2^*$ generated by $(g_1, g_2^{\gamma_1})$, independently of the committed value. Extraction: clear from the statement.

Computational Binding: assume that the adversary outputs $(\boldsymbol{a}, r_a)$ and $(\boldsymbol{b}, r_b)$ with $(\boldsymbol{a}, r_a) \neq (\boldsymbol{b}, r_b)$, such that $d(X) := (r_a P_0(X) + \sum_{i=1}^n a_i P_i(X)) - (r_b P_0(X) + \sum_{i=1}^n b_i P_i(X))$ has a root at σ. If the adversary is successful, then $d(X) \in \mathbb{Z}_p[X]$ is a non-trivial polynomial. Since the coefficients of d are known, we can use an efficient polynomial factorization algorithm [LLL82,vHN10] to compute all roots r_i of $d(X)$. One of these roots has to be equal to σ. One can establish which one by comparing each (say) $g_1^{P_1(r_i)}$ to the element $g_1^{P_1(\sigma)}$ given in the CRS. We note that $g_1^{P_1(r_i)}$ is computed from g_1 (this is why we need that crs always contains g_1), the coefficients of $P_1(X)$, and r_i. □

Another common methodology in such proofs is to use the Schwartz-Zippel lemma [Sch80]. However, the Schwartz-Zippel lemma establishes binding only under a decisional assumption, while using polynomial factorization enables us to establish binding under a computational assumption.

See App. A for some history of this commitment scheme. In the rest of this paper, we use concrete polynomials to instantiate the commitment scheme of Def. 1.

Definition 2 (Interpolating Commitment Scheme). *The interpolating commitment scheme is a $\{P_i\}$-commitment scheme, instantiated with the polynomials $P_0(X) = Z(X)$ and $P_i(X) = \ell_i(X)$ for $i \in [1 \mathinner{..} n]$ over $\mathbb{Z}_p[X]$, defined as follows. Assume n is a power of two, and $\omega_i = \omega^{i-1}$ for $i \in \{1, \ldots, n\}$, where ω is the n-th primitive root of unity modulo p (this speeds up some of the arithmetic). Then,*

- *$Z(X) := \prod_{i=1}^n (X - \omega_i) = X^n - 1$ is the unique degree n monic polynomial, such that $Z(\omega_i) = 0$ for all $i \in \{1, \ldots, n\}$.*
- *$\ell_i(X) := \prod_{j \neq i} \frac{X - \omega_j}{\omega_i - \omega_j}$ is the unique degree $n - 1$ polynomial, s.t. $\ell_i(\omega_i) = 1$ and $\ell_i(\omega_j) = 0$ for $j \neq i$.*

Here, $\mathcal{F}_{\mathsf{com},1} = \{Z(X)\} \cup \{\ell_i(X)\}_{i=1}^n$.

Clearly, ℓ_i is the ith Lagrange basis polynomial, and thus $L_{\boldsymbol{a}}(X) = \sum_{i=1}^n a_i \ell_i(X)$ is the interpolating (Lagrange) polynomial of $\boldsymbol{a}$ at points ω_i, with $L_{\boldsymbol{a}}(\omega_i) = a_i$, and can thus be computed by executing an inverse Fast Fourier Transform [GG03]. Moreover, $(\ell_i(\omega_j))_{j=1}^n = \boldsymbol{e_i}$ and $(Z(\omega_j))_{j=1}^n = \boldsymbol{0}_n$.

Remark 1. Since in the case of the interpolating commitment scheme, $\mathcal{F}_{\mathsf{com},1}$ consists of $n + 1$ linearly independent degree-$(\leq n)$ polynomials, it is a basis of the set of degree-$(\leq n)$ polynomials and thus $g_1^{\mathcal{F}_{\mathsf{com},1}(\sigma)}$ can be efficiently computed from $(g_1^{\sigma^i})_{i=0}^n$ and vice versa. Hence, $\mathcal{F}_{\mathsf{com},1}$ can be replaced with $\{X^i\}_{i=0}^n$ in all underlying assumptions. In particular, defining $\mathcal{F}^*_{\mathsf{com},1} = \{X^i\}_{i=0}^n$ and $\mathcal{F}^*_{\mathsf{com},1} = Y_1 \mathcal{F}^*_{\mathsf{com},1} = \{Y_1 X^i\}_{i=0}^n$, $(\mathcal{F}_{\mathsf{com},1}, \mathcal{F}_{\mathsf{com},2})$-PDL is equivalent to $(\mathcal{F}^*_{\mathsf{com},1}, \mathcal{F}^*_{\mathsf{com},2})$-PDL, and similarly with $(\mathcal{F}_{\mathsf{com},1}, \ldots)$-PKE.

In the case of the interpolating commitment scheme, since $1 \in \mathit{span}(\mathcal{F}_{\mathsf{com},1})$, $g_1 \in \mathsf{gk} \cap g_1^{\mathcal{F}_{\mathsf{com},1}}$, and thus one can shorten the CRS by one element. This is not true in general.

When ω_i are arbitrary, polynomial interpolation takes time $\Theta(n \log^2 n)$ [GG03]. This slows down the prover computation to $\Theta(n \log^2 n)$ non-cryptographic operations in the new adaptive Subset-Sum argument, and to $\Theta(n \log^3 n)$ in a QAP-based adaptive Circuit-SAT argument. For the existence of the n-th primitive root of unity modulo p it suffices that $(n+1) \mid (p-1)$. One can use the Cocks-Pinch method [BSS05] to construct a corresponding pairing-friendly curve. Arithmetic on such a curve is about 3 times slower than on curves without specific requirements on p.

It is easy to see that the polynomials $Z(X)$ and $\ell_i(X)$ they satisfy the requirements of Thm. 1. In fact, given Def. 1 and the statement of Thm. 1, this choice is very natural: $\ell_i(X)$ interpolate linearly independent vectors (and thus are linearly independent; in fact, they constitute a basis), and the choice to interpolate unit vectors is the conceptually clearest way of choosing $\ell_i(X)$. Another natural choice of independent polynomials is to set $P_i(X) = X^i$ as in [Gro10], but as known from the previous work, that choice results in much less efficient zero knowledge arguments.

4 Preliminaries: Zero Knowledge

We refer to App. C for an informal motivation of NIZK arguments, known impossibility results, and an explanation of why the CRS model and knowledge assumptions are needed.

An NIZK argument for a language $\mathcal{L}$ consists of three algorithms, gencrs, pro and ver. The CRS generation algorithm gencrs takes as input 1^κ (and possibly some other, public, language-dependent information like the input length) and outputs the prover CRS crs_p, the verifier CRS crs_v, and a trapdoor td. The distinction between crs_p and crs_v is not important for security, but in many applications crs_v is much shorter. The prover pro takes as an input crs_p together with a statement u and a witness w, and outputs an argument π. The verifier ver takes as an input crs_v, a statement u, and an argument π, and either accepts or rejects.

Some of the expected properties of an argument are: (i) *perfect completeness* (honest verifier always accepts honest prover's argument), (ii) *perfect witness-indistinguishability* (argument distributions corresponding to all allowable witnesses are equal), (iii) *perfect zero knowledge* (there exists an efficient simulator that can, given u, $(\mathsf{crs}_p, \mathsf{crs}_v)$ and td, output an argument that comes from the same distribution as the argument produced by the prover), (iv) *adaptive computational soundness* (if $u \notin \mathcal{L}$, then an arbitrary non-uniform probabilistic polynomial time prover has negligible success in creating a satisfying argument), and (v) *adaptive computational culpable soundness* [GOS12] (informally, if $u \notin \mathcal{L}$, then an arbitrary non-uniform probabilistic polynomial time prover has negligible success in creating a satisfying argument together with a witness that $u \notin \mathcal{L}$). An argument is an argument of knowledge, if from an accepting argument it follows that the prover knows the witness. For the sake of completeness, we give formal definitions in App. C.

5 Product Argument

Here and in what follows, $\boldsymbol{a} \circ \boldsymbol{b}$ denotes the Hadamard (i.e., element-wise) product of two vectors, with $(\boldsymbol{a} \circ \boldsymbol{b})_i = a_i b_i$. In a *product argument* [Gro10], the prover aims to convince the verifier that she knows how to open three commitments (A, A^{γ_1}), (B, B^{γ_1}), and (C, C^{γ_1}) to vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ correspondingly (together with the used randomizers), such that $\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{c}$. (Here, we assume the use of a polynomial commitment scheme with knowledge secret γ_1.) Thus, here we have the language $\mathcal{L}_n^\times = \{(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{a} \circ \boldsymbol{b}) \in (\mathbb{Z}_p^n)^3\}$.

First, we will follow the line of thought of [GGPR13] (but using the matrix notation of [Lip13]) to derive a new product argument together with a supporting trapdoor commitment scheme; the latter happens to be the interpolating commitment scheme. After that, we will give a full description of the argument together with a discussion of its security and efficiency.

Let I_n be the $n \times n$ identity matrix and let $\mathbf{1}_n$ be the n-dimensional all-one vector. Assume that $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{Z}_p^n$. Clearly, $\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{c}$ iff

$$(\boldsymbol{a}^\top I_n) \circ (\boldsymbol{b}^\top I_n) - (\boldsymbol{c}^\top I_n) \circ (\mathbf{1}_n^\top I_n) = \mathbf{0}_n \enspace . \tag{3}$$

Following the terminology of [Lip13], Eq. (3) describes a (non-polynomial) *quadratic arithmetic program* [GGPR13] for the arithmetic circuit $\mathcal{C}$, consisting of n parallel multiplication gates, that on given inputs $\boldsymbol{a}$ and $\boldsymbol{b}$ returns $\boldsymbol{c} = \boldsymbol{a} \circ \boldsymbol{b}$ as the output. Importantly, in Eq. (3) we use the same matrix, I_n, in the multiplications $\boldsymbol{a}^\top I_n$, $\boldsymbol{b}^\top I_n$, and $\boldsymbol{c}^\top I_n$. Briefly, this will mean that the prover commits to all three vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{a} \circ \boldsymbol{b}$ by using the interpolating commitment scheme.

Next, from Eq. (3) we obtain a *polynomial quadratic arithmetic program* for $\mathcal{C}$ as in [Lip13]. (The original approach of [GGPR13] omitted the intermediate derivation of the non-polynomial QAP.) Fix n different values $(\omega_1, \dots, \omega_n)$ as in Def. 2. Let $\ell_i(X)$ be again the ith Lagrange basis polynomial. Clearly, Eq. (3) is equivalent to the existence of a degree $2n - 2$ polynomial $Q(X) := Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X)$, such that

(i) $Q(X) = A(X)B(X) - C(X)$, where $A(X)$, $B(X)$ and $C(X)$ belong to the span of $\{\ell_i(X)\}$ (i.e., are degree-$(n-1)$ interpolating polynomials of *some* vectors $\boldsymbol{a}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, correspondingly), and

(ii) $Q(X)$ evaluates to 0 at all n values ω_i.

In the Subset-Sum argument, we will guarantee (i) by using a knowledge assumption. We assume its truth by now. Then, for $Z(X)$ defined as in Def. 2, Eq. (3) is equivalent to $Z(X) \mid Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X)$. That is, $\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{c}$ iff there exists a degree $(2n - 2) - n = n - 2$ polynomial $\pi(X)$, such that

$$\pi(X) \cdot Z(X) = Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X) \enspace . \tag{4}$$

Clearly, a honest prover can compute $\pi(X)$ as $\pi(X) \leftarrow Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{a}\circ\boldsymbol{b}}(X)/Z(X)$.

Next, we achieve zero-knowledge as in [GGPR13] (see also [BCI$^+$13]), by introducing randomizers $r_a, r_b, r_c \leftarrow_r \mathbb{Z}_p$, and defining

$$Q_{zk}^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c};r_a,r_b,r_c}(X) := (L_{\boldsymbol{a}}(X) + r_a Z(X))\,(L_{\boldsymbol{b}}(X) + r_b Z(X)) - (L_{\boldsymbol{c}}(X) + r_c Z(X)) \enspace . \tag{5}$$

We usually assume that the superscript of Q_{zk} is clear from the context, and thus omit it.

Here, the new addends of type $r_a Z(X)$ guarantee hiding and because of their inclusion, the degree of $Q_{zk}(X)$ is $2n$. On the other hand, due to the use of $Z(X)$ in addends, $Q_{zk}(X)$ remains divisible by $Z(X)$ if and only if $\boldsymbol{c} = \boldsymbol{a} \circ \boldsymbol{b}$. Thus, $\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{c}$ if and only if

1. $Q_{zk}(X)$ can be expressed as $Q_{zk}(X) = A(X)B(X) - C(X)$ for some polynomials $A(X)$, $B(X)$ and $C(X)$ that belong to the span of $\mathcal{F}_{\mathsf{com},1}$, and
2. there exists a polynomial $\pi_{zk}(X)$, such that

$$\pi_{zk}(X) \cdot Z(X) = Q_{zk}(X) \enspace . \tag{6}$$

In this case (i.e., if $\pi_{zk}(X)$ exists), it has degree n, and can be computed as

$$\pi_{zk}(X) := Q_{zk}(X)/Z(X) \enspace . \tag{7}$$

However, $|\pi_{zk}(X)|$ is not of sublinear in n. As common in such situations, to minimize communication, we let the prover to transfer the evaluation of $\pi_{zk}(X)$ at a random secret point σ. Since σ is an unknown secret key, the prover cannot compute $\pi_{zk}(\sigma)$. Instead, he computes $\pi_\times := g_1^{\pi_{zk}(\sigma)}$, using the values $g_1^{\sigma^i}$ (given in the CRS) and the coefficients π_i of $\pi_{zk}(X) = \sum_{i=0}^n \pi_i X^i$ (computed as in Eq. (7)), as follows:

$$\pi_\times := g_1^{\pi_{zk}(\sigma)} \leftarrow \prod_{i=0}^{n} (g_1^{\sigma^i})^{\pi_i} \quad . \tag{8}$$

Product Argument: Details. We now give a detailed description of the new product argument. Let com be the interpolating commitment scheme. Note that $\mathsf{com}(\mathsf{ck}; \mathbf{1_n}; 0) = (g_1, g_2^{\gamma_1})$.

CRS generation $\mathsf{gencrs}_\times(1^\kappa, n)$**:** Let $\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, \hat{e}) \leftarrow \mathsf{genbp}(1^\kappa)$ and $g_2 \leftarrow_r \mathbb{G}_2^*$. Generate $(\sigma, \gamma_1) \leftarrow_r \mathbb{Z}_p^2$ with $Z(\sigma) \neq 0$ and $\gamma_1 \neq 0$. Set $\mathsf{crs}_p = \mathsf{ck} \leftarrow (\mathsf{gk}; (g_1, g_2^{\gamma_1})^{\mathcal{F}_{\mathsf{com},1}(\sigma)})$. Let $\mathsf{crs}_v \leftarrow (\mathsf{gk}; g_2^{\gamma_1 Z(\sigma)})$, and $\mathsf{td}_\times \leftarrow (\sigma, \gamma_1)$. Output $(\mathsf{crs}_\times = (\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}_\times)$.

Common inputs: $\mathsf{inp}_\times = (A_1, A_2^{\gamma_1}, B_1, B_2^{\gamma_1}, C_1, C_2^{\gamma_1})$.

Proving $\mathsf{pro}_\times(\mathsf{crs}_p; \mathsf{inp}_\times; w_\times = (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{c}, r_c))$**:** Compute $\pi_{zk}(X) = \sum_{i=0}^n \pi_i X^i$ as in Eq. (7) and $\pi_\times$ as in Eq. (8). Output $\pi_\times$.

Verification $\mathsf{ver}_\times(\mathsf{crs}_v; \mathsf{inp}_\times; \pi_\times)$**:** Verify that $\hat{e}(A_1, B_2^{\gamma_1}) = \hat{e}(g_1, C_2^{\gamma_1}) \cdot \hat{e}(\pi_\times, g_2^{\gamma_1 Z(\sigma)})$.

Inclusion of $g_2^{\gamma_1 Z(\sigma)}$ in the CRS is only needed to speed up the verification; one can clearly recompute it from ck. In particular, it suffices to take $\mathsf{crs}_\times = (\mathsf{gk}; \mathsf{ck})$. We note that here as in the shift argument of Sect. 6, validity of the commitments will be verified in the Subset-Sum argument. This is since the Subset-Sum argument uses some of the commitments in several subarguments, while it suffices to verify the validity of every commitment only once.

Security. Like other basic arguments, the product argument cannot satisfy the standard definition of soundness, see [Gro10,Lip12]. It can only satisfy culpable soundness. Its culpable soundness is sufficient for the Subset-Sum argument to be sound, due to the fact that there we use additional knowledge assumptions. Similarly, the product argument by itself is not zero-knowledge, but it is witness-indistinguishable; this suffices for the Subset-Sum argument to be zero-knowledge.

Theorem 2. *Let $n = \mathrm{poly}(\kappa)$. Let com be the interpolating commitment scheme from Def. 2. Let $\mathcal{F}_{\times,1} = \mathcal{F}_{\mathsf{com},1}$, $\mathcal{F}_{\times,2} = \mathcal{F}_{\mathsf{com},2}$, $\mathcal{F}_{\times,1}^* = \mathcal{F}_{\mathsf{com},1}^*$, and $\mathcal{F}_{\times,2}^* = \mathcal{F}_{\mathsf{com},2}^*$. The new product argument is perfectly complete and perfectly witness-indistinguishable. If genbp is $(\mathcal{F}_{\times,1}^*, \mathcal{F}_{\times,2}^*, \{\omega_i\}_{i=1}^n, 1)$-TSDH secure, then this argument is adaptively computationally culpably sound.*

Relying on the TSDH assumption is natural, since the soundness claim is about certain (non)divisibility.

Proof. Completeness: follows from the discussion in the beginning of this section. Witness-indistinguishability: since argument $\pi_\times$ that satisfies the verification equations is unique, all witnesses result in the same argument, and thus this argument is witness-indistinguishable.

Culpable soundness: Here, culpable soundness means that a non-uniform probabilistic polynomial-time adversary against this argument has a negligible chance, given $\mathsf{crs}_\times = (\mathsf{crs}_p, \mathsf{crs}_v) \leftarrow \mathsf{gencrs}_\times(1^\kappa, n)$ as an input, of outputting $\mathsf{inp}_\times = (A_1, A_2^{\gamma_1}, B_1, B_2^{\gamma_1}, C_1, C_2^{\gamma_1})$, an argument $\pi_\times$, and a witness $w_\times = (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{c}, r_c)$ with $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{Z}_p^n$ and $r_a, r_b, r_c \in \mathbb{Z}_p$, such that

(i) $\mathsf{ver}_\times(\mathsf{crs}_v; \mathsf{inp}_\times; \pi_\times)$ accepts,

(ii) $(A_1, A_2^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r_a)$, $(B_1, B_2^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{b}; r_b)$, and $(C_1, C_2^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{c}; r_c)$, and

(iii) for some $i \in [1 \mathinner{..} n]$, $a_i b_i \neq c_i$.

According to the previous discussion, if the argument is not sound then one of the following has to hold: (a) one of $L_{\boldsymbol{a}}(X)$, $L_{\boldsymbol{b}}(X)$, $L_{\boldsymbol{c}}(X)$ is not in the span of $\{\ell_i(X)\}$, or (b) $Z(X) \nmid Q_{zk}(X)$, where $Q_{zk}(X)$ is defined as in Eq. (5). Case (a) cannot happen because of (ii) above. We now show that case (b) is impossible under the $(\mathcal{F}^*_{\times,1}, \mathcal{F}^*_{\times,2}, \{\omega_i\}_{i=1}^n, 1)$-TSDH assumption.

We first recall from Sect. 2 that since $\mathcal{F}_{\mathsf{com},1}$ is a basis of the set of degree $\leq n$ polynomials, the $(\mathcal{F}^*_{\times,1}, \mathcal{F}^*_{\times,2}, \{\omega_i\}_{i=1}^n, 1)$-TSDH assumption is equal to the $(\mathcal{F}_{\times,1}, \mathcal{F}_{\times,2}, \{\omega_i\}_{i=1}^n, 1)$-TSDH assumption. This means that a correctly distributed $\mathsf{crs}_\times = (\mathsf{crs}_p, \mathsf{crs}_v) = (\mathsf{gk}; \mathsf{ck}) \in \mathsf{gencrs}_\times(1^\kappa, n)$ can be efficiently computed from an $(\mathcal{F}^*_{\times,1}, \mathcal{F}^*_{\times,2}, \{\omega_i\}_{i=1}^n, 1)$-TSDH challenge.

Assume that $\mathcal{A}_{sound}$ is an adversary against culpable soundness. We now construct the next adversary $\mathcal{A}_{tsdh}$ against $(\mathcal{F}^*_{\times,1}, \mathcal{F}^*_{\times,2}, \{\omega_i\}_{i=1}^n, 1)$-TSDH. Given an $(\mathcal{F}^*_{\times,1}, \mathcal{F}^*_{\times,2}, \{\omega_i\}_{i=1}^n, 1)$-TSDH challenge $ch = (\mathsf{gk}, (g_1^{f(\sigma)})_{f \in \mathcal{F}^*_{\times,1}}, (g_2^{\gamma_1 f(\sigma)})_{f \in \mathcal{F}^*_{\times,2}})$, $\mathcal{A}_{tsdh}$ first computes and then sends $\mathsf{crs}_\times$ to $\mathcal{A}_{sound}$. Assume $\mathcal{A}_{sound}$ returns $(\mathsf{inp}_\times, \pi_\times, w_\times)$ such that the conditions (i–iii) hold and $Z(X) \nmid Q_{zk}(X)$.

Since $Z(X) \nmid Q_{zk}(X)$, then for some $i \in [1 \mathinner{..} n]$, $(X - \omega_i) \nmid Q_{zk}(X)$. Write $Q_{zk}(X) = q(X)(X - \omega_i) + r$ for $r \in \mathbb{Z}_p^*$. Clearly, $\deg q(X) \leq 2n - 1$. Moreover, we write $q(X) = q_1(X)Z(X) + q_2(X)$ with $\deg q_i(X) \leq n - 1$. Since the verification succeeds, $\hat{e}(g_1, g_2^{\gamma_1})^{Q_{zk}(\sigma)} = \hat{e}(\pi_\times, g_2^{\gamma_1 Z(\sigma)})$, or $\hat{e}(g_1, g_2^{\gamma_1})^{q(\sigma)(\sigma - \omega_i) + r} = \hat{e}(\pi_\times, g_2^{\gamma_1 Z(\sigma)})$, or $\hat{e}(g_1, g_2^{\gamma_1})^{q(\sigma) + r/(\sigma - \omega_i)} = \hat{e}(\pi_\times, g_2^{\gamma_1 Z(\sigma)/(\sigma - \omega_i)})$, or

$$\hat{e}(g_1, g_2^{\gamma_1})^{1/(\sigma - \omega_i)} = (\hat{e}(\pi_\times, g_2^{\gamma_1 Z(\sigma)/(\sigma - \omega_i)}) / \hat{e}(g_1^{q(\sigma)}, g_2^{\gamma_1}))^{r^{-1}} \quad . \tag{9}$$

Now, $\hat{e}(g_1^{q(\sigma)}, g_2^{\gamma_1}) = \hat{e}(g_1^{q_1(\sigma)}, g_2^{\gamma_1 Z(\sigma)}) \hat{e}(g_1^{q_2(\sigma)}, g_2^{\gamma_1})$, and thus it can be computed from $((g_1^{\sigma^i})_{i=0}^{n-1}, g_2^{\gamma_1}, g_2^{\gamma_1 Z(\sigma)}) \subset \mathsf{crs}$ by using generic group operations. Moreover, $Z(X)/(X - \omega_i) = \ell_i(X) \prod_{j \neq i} (\omega_i - \omega_j)$, and thus $g_2^{\gamma_1 Z(\sigma)/(\sigma - \omega_i)}$ can be computed from $g_2^{\gamma_1 \ell_i(\sigma)}$ by using generic group operations. Hence, the right-hand side of Eq. (9) can be computed from $((g_1^{\sigma^i})_{i=0}^{n-1}, g_2^{\gamma_1}, g_2^{\gamma_1 Z(\sigma)}, (g_2^{\gamma_1 \ell_i(\sigma)})_{i=1}^n)$ (that can be computed from ch), by using generic group operations. Thus, the adversary has computed $(\omega_i, \hat{e}(g_1, g_2^{\gamma_1})^{1/(\sigma - \omega_i)})$, for $\omega_i \neq \sigma$, and thus broken the $(\mathcal{F}^*_{\times,1}, \mathcal{F}^*_{\times,2}, \{\omega_i\}_{i=1}^n, 1)$-TSDH assumption. □

Finally, in some papers like [PGHR13], one had to include elements $g_1^{\sigma^i}$ for $i \leq 2n$ to the CRS for the security reduction to TSDH to go through. The small trick of writing $q(X) = q_1(X)Z(X) + q_2(X)$ (or something similar) can also be used there to reduce the strength of the TSDH assumption.

Efficiency of Product Argument. The prover computation is dominated by the computation of

(i) one $(n+1)$-wide multi-exponentiation. By using the Pippenger's multi-exponentiation algorithm [Pip80], for large n this means approximately $n + 1$ bilinear-group multiplications. For small values of n, one can use the algorithm by Straus [Str64].

(ii) three polynomial interpolations, one polynomial multiplication, and one polynomial division to compute the coefficients of the polynomial $\pi_{zk}(X)$. Since polynomial division can be implemented as 2 polynomial multiplications (by using pre-computation and storing some extra information in the CRS, [GG03,Lip13]), this part is dominated by two inverse FFT-s and three polynomial multiplications. Other savings are possible, see App. B.

The verifier computation is dominated by 3 pairings. (We will count the cost of validity verifications separately in the Subset-Sum argument.) In the special case $C_1 = A_1$, the verification equation can be simplified to $\hat{e}(A_1, B_2^{\gamma_1} / g_2^{\gamma_1}) = \hat{e}(\pi_\times, g_2^{\gamma_1 Z(\sigma)})$, which saves one more pairing.

Excluding gk, the prover CRS consists of $2(n+1)$ group elements, while the verifier CRS consists of 1 group element. The CRS can be computed in time $\Theta(n)$, by using an algorithm from [BSCG$^+$13].

On Security Assumptions and Efficiency. Compared to [Gro10,Lip12,FLZ13], we use a stronger computational assumption (TSDH, instead of PDL) to show the culpable soundness of the argument. However, the previous papers [Gro10,Lip12,FLZ13] required $\pi_\times$ to be accompanied with a knowledge component $\pi_{\times,2}^{\eta}$ for some secret key η, and relied on a knowledge assumption — not needed in the new argument — to allow the adversary in the security reduction to obtain the coefficients π_i. The use of TSDH instead of PDL and an additional knowledge assumption shortens the argument by one group element, decreases the workload of the prover twice, and the computational complexity of the verifier from 5 to 3 pairings.

Restriction Argument. In a restriction argument [Gro10] for subset $S \subseteq [1\,..\,n]$, the prover aims to convince the verifier that $(A_1, A_2^{\gamma_1})$ commits to $\boldsymbol{a}$ such that $a_i = 0$ for $i \notin S$. In the case $S = \emptyset$, we have the *zero argument* [LZ13], where the prover tries to convince the verifier that $(A_1, A_2^{\gamma_1})$ commits to $\mathbf{0}_n$. In [Gro10], the restriction argument consists of 1 group element (the knowledge component). Based on the new product argument, we construct a new restriction argument, eliminating again the need for a knowledge assumption. We refer to App. D for further details.

6 Right Shift-by-z Argument

In a *right shift-by-z* argument [FLZ13], the prover aims to convince the verifier that for 2 commitments (A, A^{γ_1}) and (B, B^{γ_1}), he knows how to open them as $(A, A^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r_a)$ and $(B, B^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{b}; r_b)$, such that $a_i = b_{i+z}$ for $i \in [1\,..\,n-z]$ and $a_i = 0$ for $i \in [n-z+1\,..\,n]$. That is, the language $\mathcal{L}_n^{\mathsf{rsft}}$ consists of all pairs$(\boldsymbol{a}, \boldsymbol{b}) \in (\mathbb{Z}_p^n)^2$, such that $(a_n, \dots, a_1) = (0, \dots, 0, b_n, \dots, b_{1+z})$.

Here, one cannot use the same methodology as in Sect. 5, since then in this argument $\boldsymbol{a}$ and $\boldsymbol{b}$ have asymmetric roles and thus will be committed by using different commitment schemes, neither compatible with the interpolating commitment scheme. In fact, in both resulting commitment schemes one would use the same polynomials P_i but in a different shifted order.

An efficient right shift-by-z argument was described in [FLZ13]. We now reconstruct this argument so that it can be used together with the interpolating commitment scheme of Def. 2. There are several reasons why this is possible. Most importantly, the right shift argument of [FLZ13] is very efficient to start with, and its construction *almost* does not depend on the commitment scheme. Indeed, the new argument needs one non-trivial modification compared to [FLZ13]: as we will see in what follows, for the security reasons we set a certain polynomial $V(X)$ to be equal to $Z(X)$, while in [FLZ13], $V(X)$ had a different definition $V(X) = X^z$. We also slightly optimize the resulting argument; in particular, the verifier has to execute one less pairing compared to [FLZ13].

Our strategy of constructing a shift argument follows the strategy of [Gro10] and follow-up papers. We start with a fixed commitment scheme and a fixed verification equation that also contains the argument. We write the discrete logarithm of the argument (that follows from this equation) as a sum of two polynomials $F_\pi(X)$ and $F_{con}(X)$, each of which belongs to the span of a set of polynomials. The first polynomial, $F_\pi(X)$, is identically zero if and only if the prover is honest. Under the assumption that the spans of certain two polynomial sets do not intersect, this results in an efficient shift argument that is culpably sound under a knowledge assumption.

Following the blueprint of [FLZ13], assume that the verification equation is $\hat{e}(B_1\pi_1, g_2^{\gamma_1 V(\sigma)}) = \hat{e}(A_1, g_2^{\gamma_1 V(\sigma)^2})$, for $(A_1, A_2^{\gamma_1})$ and $(B_1, B_2^{\gamma_1})$ being commitments to $\boldsymbol{a}$ and $\boldsymbol{b}$ (by using the $\{P_i\}$-commitment scheme, without fixing the polynomials yet), γ_1 being a knowledge secret, and $V(X)$ being a non-zero polynomial that we will fix later. The value $(g_1^{\pi(\sigma)}, g_2^{\gamma_2\pi(\sigma)})$ corresponds to the argument, where γ_2 is another knowledge secret. Denote $r(X) := r_a P_0(X)V(X) - r_b P_0(X)$. Replacing σ with a formal variable X and taking a discrete logarithm of the verification equation,

$$\begin{aligned}\pi(X) = &\left(r_a P_0(X) + \sum_{i=1}^{n} a_i P_i(X)\right) V(X) - \left(r_b P_0(X) + \sum_{i=1}^{n} b_i P_i(X)\right) \\ =& V(X) \sum_{i=1}^{n} a_i P_i(X) - \sum_{i=1}^{n} b_i P_i(X) + r(X) \\ =& \left(\sum_{i=1}^{n-z} a_i P_i(X) + \sum_{i=n-z+1}^{n} a_i P_i(X)\right) V(X) - \left(\sum_{i=1}^{n-z} b_{i+z} P_{i+z}(X) + \sum_{i=1}^{z} b_i P_i(X)\right) + r(X) \enspace ,\end{aligned}$$

and thus $\pi(X) = F_\pi(X) + F_{con}(X)$, where

$$\begin{aligned}F_\pi(X) =& \left(\sum_{i=1}^{n-z} (a_i - b_{i+z}) P_i(X) + \sum_{i=n-z+1}^{n} a_i P_i(X)\right) V(X) \enspace , \\ F_{con}(X) =& \left(\sum_{i=z+1}^{n} b_i (P_{i-z}(X)V(X) - P_i(X)) - \sum_{i=1}^{z} b_i P_i(X)\right) + r(X) \enspace .\end{aligned}$$

Now, the prover is honest if and only if $F_\pi(X) = 0$ if and only if $\pi(X) = F_{con}(X)$, i.e., $\pi(X)$ belongs to the span of $\varPhi_{z-\mathsf{rsft}} := \{P_{i-z}(X)V(X) - P_i(X)\}_{i=z+1}^{n} \cup \{P_i(X)\}_{i=1}^{z} \cup \{P_0(X)V(X)\} \cup \{P_0(X)\}$. We guarantee this by a knowledge assumption; for this reason we will also show that $\varPhi_{z-\mathsf{rsft}}$ is linearly independent. As in the case of the product argument, we also need that A and B are actually commitments of n-dimensional vectors, i.e., we only prove culpable soundness.

For the shift argument to be sound, we need that (i) $\{P_i(X)V(X)\}_{i=1}^{n}$ is linearly independent (since $V(X)$ is non-zero, this follows from the linear independence of $\{P_i(X)\}$), and (ii) $F_\pi(X) \cap \mathit{span}(\varPhi_{z-\mathsf{rsft}}) = \emptyset$, for which it suffices that $P_k(X)V(X) \notin \mathit{span}(\varPhi_{z-\mathsf{rsft}})$ for $k \in [1 \mathinner{..} n]$. Both (i) and (ii) together guarantee that from a representation of $\pi(X)$ as an element of $\mathit{span}(\varPhi_{z-\mathsf{rsft}})$ it follows that $\boldsymbol{a}$ is a shift of $\boldsymbol{b}$.

We now show that one can use the interpolating commitment scheme of Def. 2 together with a concrete choice of $V(X)$.

Lemma 1. *For the interpolating commitment scheme of Def. 2 and $V(X) = Z(X)$, let*

$$\varPhi_{z-\mathsf{rsft}} := \{\ell_{i-z}(X)Z(X) - \ell_i(X)\}_{i=z+1}^{n} \cup \{\ell_i(X)\}_{i=1}^{z} \cup \{Z(X)^2\} \cup \{Z(X)\} \enspace .$$

It holds that $\varPhi_{z-\mathsf{rsft}}$ is linearly independent and that $\ell_k(X)Z(X) \notin \mathit{span}(\varPhi_{z-\mathsf{rsft}})$ for any $k \in [1 \mathinner{..} n]$.

Proof. We will prove the second claim; the first claim can be proven analogously (see, e.g., the proof of Lem. 2). Assume that for some $k \in [1 \mathinner{..} n]$, $\ell_k(X)Z(X) \in \mathit{span}(\varPhi_{z-\mathsf{rsft}})$. Thus, there exist $\boldsymbol{a} \in \mathbb{Z}_p^n$, $\boldsymbol{b} \in \mathbb{Z}_p^n$, $c \in \mathbb{Z}_p$, and $d \in \mathbb{Z}_p$, s.t. $\ell_k(X)Z(X) = \sum_{i=z+1}^{n} a_i (\ell_{i-z}(X)Z(X) - \ell_i(X)) + \sum_{i=1}^{z} b_i \ell_i(X) + cZ(X)^2 + dZ(X)$. But then the left hand side and the right hand side polynomials must also agree at ω_j, for $j \in [1 \mathinner{..} n]$. Therefore, due to the definition of $\ell_i(X)$ and $Z(X)$, $\mathbf{0}_n = -\sum_{i=z+1}^{n} a_i \boldsymbol{e_i} + \sum_{i=1}^{z} b_i \boldsymbol{e_i}$. The latter is only possible if $a_i = b_j = 0$ for all $i \in [z+1 \mathinner{..} n]$ and $j \in [1 \mathinner{..} z]$. Since $\ell_k(X)Z(X) \neq cZ(X)^2 + dZ(X)$ for constant c and d, this finishes the proof. □

Since culpable soundness of the new shift argument relies on $\pi(X)$ belonging to a certain span, similarly to [FLZ13], we will use an additional knowledge assumption. That is, for its culpable soundness it is necessary that the adversary also outputs a witness that $\pi(X) = F_{con}(X)$ belongs to the span of $\Phi_{z-\mathsf{rsft}}$.

Similarly to the product argument, the shift argument does not contain the polynomial $\pi(X) = F_{con}(X)$ itself, but the value $(g_1, g_2^{\gamma_2})^{\pi(\sigma)}$ for random σ and a knowledge secret γ_2 (necessary due to the use of knowledge assumption), computed as

$$\pi_{\mathsf{rsft}} \leftarrow (\pi_1, \pi_2^{\gamma_2}) = (g_1, g_2^{\gamma_2})^{\pi(\sigma)} = \prod_{i=z+1}^{n} \left((g_1, g_2^{\gamma_2})^{\ell_{i-z}(\sigma)Z(\sigma)-\ell_i(\sigma)}\right)^{b_i} \cdot \prod_{i=1}^{z} \left((g_1, g_2^{\gamma_2})^{\ell_i(\sigma)}\right)^{-b_i} \cdot \tag{10}$$
$$\left((g_1, g_2^{\gamma_2})^{Z(\sigma)^2}\right)^{r_a} \cdot \left((g_1, g_2^{\gamma_2})^{Z(\sigma)}\right)^{-r_b} .$$

We are now ready to state the new right-shift-by-z argument:

CRS generation $\mathsf{gencrs}_{\mathsf{rsft}}(1^\kappa, n)$**:** Let $\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, \hat{e}) \leftarrow \mathsf{genbp}(1^\kappa)$, and $g_2 \leftarrow_r \mathbb{G}_2^*$. Generate $(\sigma, \gamma_1, \gamma_2) \leftarrow \mathbb{Z}_p^3$ with $Z(\sigma) \neq 0$ and $\gamma_1 \neq 0$. Set $\mathsf{ck} \leftarrow (\mathsf{gk}; (g_1, g_2^{\gamma_1})^{\mathcal{F}_{\mathsf{com},1}(\sigma)}) = \mathsf{gencom}(1^\kappa, n)$. Let $\mathsf{crs}_p \leftarrow (\mathsf{gk}; \mathsf{ck}, (g_1, g_2^{\gamma_2})^{\Phi_{z-\mathsf{rsft}}(\sigma)})$, $\mathsf{crs}_v \leftarrow (\mathsf{gk}; (g_1, g_2^{\gamma_2})^{Z(\sigma)}, g_2^{\gamma_2 Z(\sigma)^2})$. Set $\mathsf{td}_{\mathsf{rsft}} \leftarrow (\sigma, \gamma_1, \gamma_2)$. Return $(\mathsf{crs}_{\mathsf{rsft}} = (\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}_{\mathsf{rsft}})$.

Common inputs: $\mathsf{inp}_{\mathsf{rsft}} = (A_1, A_2^{\gamma_1}, B_1, B_2^{\gamma_1})$.

Proving $\mathsf{pro}_{\mathsf{rsft}}(\mathsf{crs}_p; \mathsf{inp}_{\mathsf{rsft}}; \boldsymbol{a}, r_a, \boldsymbol{b}, r_b)$**:** compute $\pi_{\mathsf{rsft}} \leftarrow (\pi_1, \pi_2^{\gamma_2})$ as in Eq. (10). Return π_{rsft}.

Verification $\mathsf{ver}_{\mathsf{rsft}}(\mathsf{crs}_v; \mathsf{inp}_{\mathsf{rsft}}; \pi_{\mathsf{rsft}} = (\pi_1, \pi_2^{\gamma_2}))$**:** check that $\hat{e}(\pi_1, g_2^{\gamma_2 Z(\sigma)}) = \hat{e}(g_1^{Z(\sigma)}, \pi_2^{\gamma_2})$ and $\hat{e}(g_1^{Z(\sigma)}, B_2^{\gamma_2} \pi_2^{\gamma_2}) = \hat{e}(B_1 \pi_1, g_2^{\gamma_2 Z(\sigma)}) = \hat{e}(A_1, g_2^{\gamma_2 Z(\sigma)^2})$.

Let $\mathcal{F}_{z-\mathsf{rsft},1} = \mathcal{F}_{\mathsf{com},1} \cup \Phi_{z-\mathsf{rsft}}$ and $\mathcal{F}_{z-\mathsf{rsft},2} = \mathcal{F}_{\mathsf{com},2} \cup Y_2 \Phi_{z-\mathsf{rsft}}$. Clearly, here $\mathsf{crs}_{\mathsf{rsft}} = (\mathsf{gk}; g_1^{\mathcal{F}_{z-\mathsf{rsft},1}(\sigma)}, g_2^{\gamma_1 \mathcal{F}_{\mathsf{com},1}(\sigma)}, g_2^{\gamma_2 \Phi_{z-\mathsf{rsft}}(\sigma)}) = (\mathsf{gk}; g_1^{\mathcal{F}_{z-\mathsf{rsft},1}(\sigma)}, g_2^{\mathcal{F}_{z-\mathsf{rsft},2}(\sigma, \gamma_1, \gamma_2)})$. This holds since crs_v can be recomputed from crs_p.

Theorem 3. *Let $n = \mathrm{poly}(\kappa)$. Let* com *be the interpolating commitment scheme. The shift argument of the current section is perfectly complete and perfectly witness-indistinguishable. Let $\Phi_{z-\mathsf{rsft}}$ be as in Lem. 1. If* genbp *is $(\mathcal{F}_{z-\mathsf{rsft},1}, \mathcal{F}_{z-\mathsf{rsft},2})$-PDL secure and $(\Phi_{1-\mathsf{rsft}}, \emptyset, \emptyset, 2)$-PKE secure, then the shift argument is adaptively computationally culpably sound.*

The proof of this theorem is given in App. E.

Efficiency of Shift Argument. The prover computation is dominated by two $(n+2)$-wide multi-exponentiations. This time, there is no need for polynomial interpolation, multiplication or division. The communication is 2 group elements. The verifier computation is dominated by 5 pairings. Apart from gk, the prover CRS contains $2((n+1)+(n+2)) = 4n+6$ group elements, and the verifier CRS contains 3 group elements.

Simplifying the PDL Assumption. We will now show that the underlying PDL assumption is implied by a simpler PDL assumption. We first show that if $z = 1$, then the assumptions are actually equivalent. For this, note that $\mathcal{F}_{1-\mathsf{rsft},1} = \mathcal{F}_{\mathsf{com},1} \cup \Phi_{1-\mathsf{rsft}} := \{\ell_{i-1}(X)Z(X) - \ell_i(X)\}_{i=2}^{n} \cup \{\ell_i(X)\}_{i=1}^{n} \cup \{Z(X)^2\} \cup \{Z(X)\}$. See App. F for a proof of the next lemma.

Lemma 2. *Define $\mathcal{F}_{1-\mathsf{rsft},1}^* := \{X^i\}_{i=0}^{2n}$ and $\mathcal{F}_{1-\mathsf{rsft},2}^* := Y_1 \mathcal{F}_{1-\mathsf{rsft},1}^* = \{Y_1 X^i\}_{i=0}^{2n}$. The $(\mathcal{F}_{1-\mathsf{rsft},1}, \mathcal{F}_{1-\mathsf{rsft},2}^1)$-PDL assumption is equal to the $(\mathcal{F}_{1-\mathsf{rsft},1}^*, \mathcal{F}_{1-\mathsf{rsft},2}^*)$-PDL assumption.*

Corollary 1. *In Thm. 3, one can replace the* $(\mathcal{F}_{1-\mathsf{rsft},1}, \mathcal{F}^{1}_{1-\mathsf{rsft},2})$*-PDL assumption with the* $(\mathcal{F}^{*}_{1-\mathsf{rsft},1}, \mathcal{F}^{*}_{1-\mathsf{rsft},2})$*-PDL assumption.*

In general, since $\mathcal{F}_{z-\mathsf{rsft},1}$ consists of degree $\leq 2n$ polynomials, $(\mathcal{F}_{z-\mathsf{rsft},1}, \mathcal{F}_{z-\mathsf{rsft},2})$-PDL is implied by $(\mathcal{F}^{*}_{1-\mathsf{rsft},1}, \mathcal{F}^{*}_{1-\mathsf{rsft},2})$-PDL. However, $\mathcal{F}_{z-\mathsf{rsft},1}$ and $\mathcal{F}_{z-\mathsf{rsft},2}$ have $(n{+}1){+}(n{+}2){-}(z{+}1) = 2n{+}2{-}z$ elements, since $|\mathcal{F}_{\mathsf{com},1} \cap \varPhi_{z-\mathsf{rsft}}| = z + 1$. Thus, $(\mathcal{F}_{(z+1)-\mathsf{rsft},1}, \mathcal{F}_{(z+1)-\mathsf{rsft},2})$-PDL is a presumably weaker assumption than $(\mathcal{F}_{z-\mathsf{rsft},1}, \mathcal{F}_{z-\mathsf{rsft},2})$-PDL for any z.

7 Subset-Sum Argument

For a fixed n, the NP-complete language Subset-Sum is usually defined as the language $\mathcal{L}_{n}^{\textsc{Subset-Sum}}$ of tuples $(\boldsymbol{S} = (S_1, \dots, S_n), s)$ such that there exists a vector $\boldsymbol{b} \in \{0,1\}^n$ with $\sum_{i=1}^{n} S_i b_i = s$. Thus, in a Subset-Sum argument, the prover aims to convince the verifier that he knows how to open commitment $(B_1, B_2^{\gamma_1})$ to a vector $\boldsymbol{b} \in \{0,1\}^n$, such that $\sum_{i=1}^{n} S_i b_i = s$.

Next, we show that by using the new product and shift arguments, one can design a computationally efficient adaptive short Subset-Sum NIZK argument.

Construction. To prove that $(\boldsymbol{S}, s) \in$ Subset-Sum, we do the following. The CRS generation $\mathsf{gencrs}_{\mathsf{ssum}}$ invokes the CRS generations of the commitment scheme, the product argument and the shift argument, sharing the same gk and trapdoor $\mathsf{td} = (\sigma, \gamma_1, \gamma_2)$ between the different invocations.

The prover does the following (further explanations are given in the completeness proof):

```
Let $\boldsymbol{b} \in \{0,1\}^n$ be such that $\sum_{i=1}^{n} S_i b_i = s$.
Construct a product argument $\pi_1$ to show that $\boldsymbol{b} \circ \boldsymbol{b} = \boldsymbol{b}$.
Let $(C_1, C_2^{\gamma_1})$ be a commitment to $\boldsymbol{c} \leftarrow \boldsymbol{S} \circ \boldsymbol{b}$.
Construct a product argument $\pi_2$ to show that $\boldsymbol{c} = \boldsymbol{S} \circ \boldsymbol{b}$.
Let $(D_1, D_2^{\gamma_1})$ be a commitment to $\boldsymbol{d}$, where $d_i = \sum_{j \geq i} c_j$.
Construct a shift argument $(\pi_{31}, \pi_{32}^{\gamma_2})$ to show that $\boldsymbol{d} - \boldsymbol{c}$ is a right shift-by-1 of $\boldsymbol{d}$.
Construct a product argument $\pi_4$ to show that $\boldsymbol{e_1} \circ (\boldsymbol{d} - s\boldsymbol{e_1}) = \boldsymbol{0}_n$.
Output $\pi_{\mathsf{ssum}} = (B_1, B_2^{\gamma_1}, C_1, C_2^{\gamma_1}, D_1, D_2^{\gamma_1}, \pi_1, \pi_2, \pi_{31}, \pi_{32}^{\gamma_2}, \pi_4)$.
```

The vector $\boldsymbol{d}$ is called either a *vector scan*, an *all-prefix-sums* or a *prefix-sum* of $\boldsymbol{c}$ [Ble90], and $(\pi_{31}, \pi_{32}^{\gamma_2})$ can be thought of as a *scan argument* [FLZ13] that $\boldsymbol{d}$ is a correct scan of $\boldsymbol{c}$.

After receiving π_{ssum}, the verifier checks the validity of three commitments $(B_1, B_2^{\gamma_1})$, $(C_1, C_2^{\gamma_1})$ (for this, the verifier has to also compute a commitment to $\boldsymbol{S}$), and $(D_1, D_2^{\gamma_1})$, and then verifies four basic arguments.

Let $\mathcal{F}_{\mathsf{ssum},1} = \mathcal{F}_{\times,1} \cup \varPhi_{1-\mathsf{rsft}}$ and $\mathcal{F}_{\mathsf{ssum},2} = \mathcal{F}_{\times,2} \cup Y_2\varPhi_{1-\mathsf{rsft}}$. Clearly, here it suffices to take $\mathsf{crs} = (\mathsf{gk}; g_1^{\mathcal{F}_{\mathsf{ssum},1}(\sigma)}, g_2^{\mathcal{F}_{\mathsf{ssum},2}(\sigma,\gamma_1,\gamma_2)})$. See App. G for a proof of the following theorem.

Theorem 4. *Let* $n = \mathrm{poly}(\kappa)$*, and let* com *be the interpolating commitment scheme. The new* Subset-Sum *argument is perfectly complete and perfectly zero-knowledge. It is adaptively computationally sound and an argument of knowledge if* genbp *is* $(\mathcal{F}^{*}_{\times,1} = \{X^i\}_{i=0}^{n}, \mathcal{F}^{*}_{\times,2} = \{Y_1 X^i\}_{i=0}^{n}, \{\omega_i\}_{i=1}^{n}, 1)$*-TSDH,* $(\mathcal{F}^{*}_{1-\mathsf{rsft},1} = \{X^i\}_{i=0}^{2n}, \mathcal{F}^{*}_{1-\mathsf{rsft},2} = \{Y_2 X^i\}_{i=0}^{2n})$*-PDL,* $(\mathcal{F}^{*}_{\mathsf{com},1} = \{X^i\}_{i=0}^{n}, \mathcal{F}_{\mathsf{ssum},1} \setminus \mathcal{F}_{\mathsf{com},1}, Y_2\varPhi_{1-\mathsf{rsft}}, 1)$*-PKE, and* $(\varPhi_{1-\mathsf{rsft}}, \mathcal{F}_{\times,1}, \mathcal{F}_{\times,2}, 2)$*-PKE secure.*

Efficiency. The prover computation is dominated by three commitments and the application of three product arguments and one shift argument, that is, by $\Theta(n \log n)$ non-cryptographic operations and $\Theta(n)$ cryptographic operations. The latter is dominated by 11 ($\approx n$)-wide multi-exponentiations (2 in each commitment and the shift argument, and 1 in each product argument). The argument size is constant (11 group elements), and the verifier computation is dominated by two $(n+1)$-wide multi-exponentiations (needed to once commit to $\boldsymbol{S}$; this can often be pre-computed) and 18 pairings (3 pairings to verify π_2, 2 pairings to verify each of the other product arguments, 5 pairings to verify the shift argument, and 6 pairings to verify the validity of 3 commitments). As always, multi-exponentiation can be significantly sped up by using algorithms from [Str64,Pip80].

Weaker Assumptions. In the first three assumptions of Thm. 4, the first set of polynomials ($\mathcal{F}^*_{\times,1}$, $\mathcal{F}^*_{1-\mathsf{rsft}}$, $\mathcal{F}^*_{\mathsf{com},1}$, resp.) is equal to $\{X^i\}_{i=0}^d$ for some d. This is not the case with the fourth assumption. In App. H, we modify the shift argument (and thus also the Subset-Sum argument) so as to bring the fourth assumption into line. However, this results in the use of a presumably stronger computational assumption.

8 Other Arguments

As shown in [FLZ13], arguments for other interesting languages can be constructed, given efficient product and shift arguments. This includes NP-complete languages like Partition but also range argument. One can plug in the interpolating commitment scheme and the new product argument to speed up corresponding arguments. For example, the range argument in [FLZ13] consists of product and shift arguments. In fact, it looks very similar to the new Subset-Sum argument, except that one has additionally commit to a value $a \in [0\mathinner{..}H]$, use a specific sparse $\boldsymbol{S} = \boldsymbol{H}$ with $H_i = \lfloor (H + 2^{i-1})/2^i \rfloor$ [LAN02], and prove that $a = \sum_{i=1}^n H_i b_i$ for a committed a. Since here $\boldsymbol{H}$ does not depend on instance, the verifier computation is $\Theta(1)$. See App. I for a full description of the corresponding range argument, and App. J for a discussion of other NP-complete languages.

When instantiated with the interpolating commitment scheme, all such arguments will have prover computation dominated by $\Theta(n \log n)$ non-cryptographic operations and $\Theta(n)$ cryptographic operations. Here, n is some language-dependent parameter (e.g., the size of the integer set in Subset-Sum). It is unknown how to achieve similar efficiency by using any other techniques.

Recently, Lipmaa [Lip14] designed a permutation argument that uses $\Theta(\log n)$ product and shift arguments. By using the permutation argument of [Lip14] and the framework of [Gro10], from this and the new arguments of the current paper one can construct an adaptive Circuit-SAT argument with complexity parameters as stated in Tbl. 1. See [Lip14] for more discussion.

Acknowledgments. We would like to thank Paulo Barreto and Prastudy Fauzi for useful comments. The author was supported by the Estonian Research Council, and European Union through the European Regional Development Fund.

References

BB04. Dan Boneh and Xavier Boyen. Secure Identity Based Encryption Without Random Oracles. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 443–459, Santa Barbara, USA, August 15–19, 2004. Springer, Heidelberg.

BB08. Dan Boneh and Xavier Boyen. Short Signatures Without Random Oracles and the SDH Assumption in Bilinear Groups. *Journal of Cryptology*, 21(2):149–177, 2008.

BBG05. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. Hierarchical Identity Based Encryption with Constant Size Ciphertext. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 440–456, Aarhus, Denmark, May 22–26, 2005. Springer, Heidelberg.

BCCT13. Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. Recursive Composition and Bootstrapping for SNARKs and Proof-Carrying Data. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *STOC 2013*, pages 241–250, Palo Alto, CA, USA, June 1–4, 2013. ACM Press.

BCI$^+$13. Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct Non-interactive Arguments via Linear Interactive Proofs. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 315–333, Tokyo, Japan, March 3–6, 2013. Springer, Heidelberg.

BCPR14. Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. On the Existence of Extractable One-Way Functions. In David Shmoys, editor, *STOC 2014*, pages 505–514, New York, NY, USA, May 31 – June 3, 2014. ACM Press.

Beh46. Felix A. Behrend. On the Sets of Integers Which Contain No Three in Arithmetic Progression. *Proceedings of the National Academy of Sciences*, 32(12):331–332, December 1946.

BF01. Dan Boneh and Matthew K. Franklin. Identity-Based Encryption from the Weil Pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229, Santa Barbara, USA, August 19–23, 2001. Springer, Heidelberg.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-Interactive Zero-Knowledge and Its Applications. In *STOC 1988*, pages 103–112, Chicago, Illinois, USA, May 2–4, 1988. ACM Press.

Ble90. Guy Blelloch. *Vector Models for Data-Parallel Computing*. MIT Press, 1990.

Blo14. Thomas F. Bloom. A Quantitative Improvement for Roth's Theorem on Arithmetic Progressions. Technical Report arXiv:1405.5800, arXiv.org, May 22 2014. Available at `http://arxiv.org/abs/1405.5800`.

BSCG$^+$13. Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge. In Ran Canetti and Juan Garay, editors, *CRYPTO (2) 2013*, volume 8043 of *LNCS*, pages 90–108, Santa Barbara, California, USA, August 18–22, 2013. Springer, Heidelberg.

BSCTV14. Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Scalable Zero Knowledge via Cycles of Elliptic Curves. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO (2) 2014*, volume 8617 of *LNCS*, pages 276–294, Santa Barbara, California, USA, August 17–21, 2014. Springer, Heidelberg.

BSS05. Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart, editors. *Advances in Elliptic Curves in Cryptography*. London Mathematical Society Lecture Note Series. Cambridge Univ Press, 2 edition, April 2005.

CGH98. Ran Canetti, Oded Goldreich, and Shai Halevi. The Random Oracle Methodology, Revisited. In Jeffrey Scott Vitter, editor, *STOC 1998*, pages 209–218, Dallas, Texas, USA, May 23–26, 1998.

CL07. Ernie S. Croot and Vsevolod F. Lev. *Open Problems in Additive Combinatorics*, volume 43 of *CRM Proc. Lecture Notes*, pages 207–233. Amer. Math. Soc., 2007. Updated version (2011) available at `http://people.math.gatech.edu/~ecroot/E2S-01-11.pdf`.

CLs10. Rafik Chaabouni, Helger Lipmaa, and abhi shelat. Additive Combinatorics and Discrete Logarithm Based Range Protocols. In Ron Steinfeld and Philip Hawkes, editors, *ACISP 2010*, volume 6168 of *LNCS*, pages 336–351, Sydney, Australia, July 5–7, 2010. Springer, Heidelberg.

CLZ12. Rafik Chaabouni, Helger Lipmaa, and Bingsheng Zhang. A Non-Interactive Range Proof with Constant Communication. In Angelos Keromytis, editor, *FC 2012*, volume 7397 of *LNCS*, pages 179–199, Bonaire, The Netherlands, February 27–March 2, 2012. Springer, Heidelberg.

Dam91. Ivan Damgård. Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In Joan Feigenbaum, editor, *CRYPTO 1991*, volume 576 of *LNCS*, pages 445–456, Santa Barbara, California, USA, August 11–15, 1991. Springer, Heidelberg, 1992.

DFGK14. George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square Span Programs with Applications to Succinct NIZK Arguments. In Palash Sarkar and Tetsu Iwata, editors, *ASIACRYPT 2014*, volume ? of *LNCS*, pages ?–?, Kaohsiung, Taiwan, December 7–11, 2014. Springer, Heidelberg.

Elk11. Michael Elkin. An Improved Construction of Progression-Free Sets. *Israel J. of Math.*, 184:93–128, 2011.

ET36. Paul Erdős and Paul Turán. On Some Sequences of Integers. *J. London Math. Soc.*, 11(4):261–263, 1936.

FLZ13. Prastudy Fauzi, Helger Lipmaa, and Bingsheng Zhang. Efficient Modular NIZK Arguments from Shift and Product. In Michel Abdalla, Cristina Nita-Rotaru, and Ricardo Dahab, editors, *CANS 2013*, volume 8257 of *LNCS*, pages 92–121, Paraty, Brazil, November 20–22, 2013. Springer, Heidelberg.

FS86. Amos Fiat and Adi Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In Andrew M. Odlyzko, editor, *CRYPTO 1986*, volume 263 of *LNCS*, pages 186–194, Santa Barbara, California, USA, 11–15 August 1986. Springer, Heidelberg, 1987.

GG03. Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2 edition, July 3, 2003.

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic Span Programs and NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645, Athens, Greece, April 26–30, 2013. Springer, Heidelberg.

GJ78. Michael R. Garey and David S. Johnson. “Strong” NP-Completeness Results: Motivation, Examples, and Implications. *Journal of the ACM*, 25(3):499–508, 1978.

GJ79. Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Series of Books in the Mathematical Sciences. W. H. Freeman, January 1979.

GJM02. Philippe Golle, Stanislaw Jarecki, and Ilya Mironov. Cryptographic Primitives Enforcing Communication and Storage Complexity. In Matt Blaze, editor, *FC 2002*, volume 2357 of *LNCS*, pages 120–135, Southhampton Beach, Bermuda, March 11–14, 2002. Springer, Heidelberg.

GK03. Shafi Goldwasser and Yael Tauman Kalai. On the (In)security of the Fiat-Shamir Paradigm. In *FOCS 2003*, pages 102–113, Cambridge, MA, USA, October 11–14, 2003. IEEE, IEEE Computer Society Press.

GL07. Jens Groth and Steve Lu. A Non-interactive Shuffle with Pairing Based Verifiability. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 51–67, Kuching, Malaysia, December 2–6, 2007. Springer, Heidelberg.

GMR85. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof-Systems. In Robert Sedgewick, editor, *STOC 1985*, pages 291–304, Providence, Rhode Island, USA, May 6–8, 1985. ACM Press.

GOS12. Jens Groth, Rafail Ostrovsky, and Amit Sahai. New Techniques for Noninteractive Zero-Knowledge. *Journal of the ACM*, 59(3), 2012.

Gro10. Jens Groth. Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340, Singapore, December 5–9, 2010. Springer, Heidelberg.

GW11. Craig Gentry and Daniel Wichs. Separating Succinct Non-Interactive Arguments from All Falsifiable Assumptions. In Salil Vadhan, editor, *STOC 2011*, pages 99–108, San Jose, California, USA, June 6–8, 2011. ACM Press.

Jou00. Antoine Joux. A One-Round Protocol for Tripartite Diffie-Hellman. In Wieb Bosma, editor, *ANTS 2000*, volume 1838 of *LNCS*, pages 385–394, Leiden, The Netherlands, 2–7 June 2000. Springer, Heidelberg.

JR13. Charanjit S. Jutla and Arnab Roy. Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013 (1)*, volume 8269 of *LNCS*, pages 1–20, Bangalore, India, December 1–5, 2013. Springer, Heidelberg.

JR14. Charanjit S. Jutla and Arnab Roy. Switching Lemma for Bilinear Tests and Constant-Size NIZK Proofs for Linear Subspaces. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO (2) 2014*, volume 8617 of *LNCS*, pages 295–312, Santa Barbara, California, USA, August 17–21, 2014. Springer, Heidelberg.

Kar72. Richard M. Karp. Reducibility Among Combinatorial Problems. In Raymond E. Miller and James W. Thatcher, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103, Thomas J. Watson Research Center, Yorktown Heights, New York, March 20–22, 1972. Plenum Press, New York.

KS08. Vladimir Kolesnikov and Thomas Schneider. A Practical Universal Circuit Construction and Secure Evaluation of Private Functions. In Gene Tsudik, editor, *FC 2008*, volume 5143 of *LNCS*, pages 83–97, Cozumel, Mexico, January 28–31, 2008. Springer, Heidelberg.

KZG10. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-Size Commitments to Polynomials and Their Applications. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 177–194, Singapore, December 5–9, 2010. Springer, Heidelberg.

LAN02. Helger Lipmaa, N. Asokan, and Valtteri Niemi. Secure Vickrey Auctions without Threshold Trust. In Matt Blaze, editor, *FC 2002*, volume 2357 of *LNCS*, pages 87–101, Southhampton Beach, Bermuda, March 11–14, 2002. Springer, Heidelberg.

Lip03. Helger Lipmaa. On Diophantine Complexity and Statistical Zero-Knowledge Arguments. In Chi Sung Laih, editor, *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 398–415, Taipei, Taiwan, November 30–December 4, 2003. Springer, Heidelberg.

Lip12. Helger Lipmaa. Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189, Taormina, Italy, March 18–21, 2012. Springer, Heidelberg.

Lip13. Helger Lipmaa. Succinct Non-Interactive Zero Knowledge Arguments from Span Programs and Linear Error-Correcting Codes. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013 (1)*, volume 8269 of *LNCS*, pages 41–60, Bangalore, India, December 1–5, 2013. Springer, Heidelberg.

Lip14. Helger Lipmaa. Efficient NIZK Arguments via Parallel Verification of Benes Networks. In Michel Abdalla and Roberto de Prisco, editors, *SCN 2014*, volume 8642 of *LNCS*, pages 416–434, Amalfi, Italy, September 3–5, 2014. Springer, Heidelberg.

LLL82. Arjen K. Lenstra, Hendrik W. Lenstra, Jr., and Laszlo Lovász. Factoring Polynomials with Rational Coefficients. *Mathematische Annalen*, 261:513–534, 1982.

LZ13. Helger Lipmaa and Bingsheng Zhang. A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. *Journal of Computer Security*, 21(5):685–719, 2013.

PGHR13. Bryan Parno, Craig Gentry, Jon Howell, and Mariana Raykova. Pinocchio: Nearly Practical Verifiable Computation. In *IEEE Symposium on Security and Privacy*, pages 238–252, San Francisco, CA, USA, May 19–22, 2013. IEEE Computer Society.

Pip80. Nicholas Pippenger. On the Evaluation of Powers and Monomials. *SIAM J. Comput.*, 9(2):230–250, 1980.

RS86. Michael O. Rabin and Jeffrey O. Shallit. Randomized Algorithms in Number Theory. *Communications in Pure and Applied Mathematics*, 39:239–256, 1986.

San11. Tom Sanders. On Roth's Theorem on Progressions. *Ann. of Math.*, 174(1):619–636, July 2011.

Sch80. Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM*, 27(4):701–717, 1980.

SOK00. Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems Based on Pairing. In *SCIS 2000*, Okinawa, Japan, 2000.

SS08. Ahmad-Reza Sadeghi and Thomas Schneider. Generalized Universal Circuits for Secure Evaluation of Private Functions with Application to Data Classification. In Pil Joong Lee and Jung Hee Cheon, editors, *ICISC 2008*, volume 5461 of *LNCS*, pages 336–353, Seoul, Korea, December 3–5, 2008. Springer, Heidelberg.

Str64. Ernst G. Straus. Addition Chains of Vectors. *American Mathematical Monthly*, 70:806–808, 1964.

TV06. Terrence Tao and Van Vu. *Additive Combinatorics.* Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2006.

Val76. Leslie G. Valiant. Universal Circuits (Preliminary Report). In *STOC 1976*, pages 196–203, Hershey, Pennsylvania, USA, May 3–5, 1976. ACM.

vHN10. Mark van Hoeij and Andrew Novocin. Gradual Sub-lattice Reduction and a New Complexity for Factoring Polynomials. In Alejandro López-Ortiz, editor, *LATIN 2010*, volume 6034 of *LNCS*, pages 539–553, Oaxaca, Mexico, April 19–23, 2010. Springer, Heidelberg.

A Commitment Scheme: History

For different special cases (e.g., $P_0(X) = 1$ and $P_i(X) = X^i$ for $i \in [1 \,..\, n]$), versions of the $\{P_i\}$-commitment scheme have been in use since at least [GJM02]. Groth [Gro10] seems to be first who used such a commitment scheme in conjunction with a knowledge assumption. In [Lip12], the authors considered the case of $P_0(X) = 1$ and $P_i(X) = X^{\lambda_i}$, $i \in [1 \,..\, n]$, with λ_i having specific properties, related to the concrete application. In the case of [Lip12], dependence on applications is not bad, since the possible "application" is one of the relatively small set of arguments (e.g., the product argument or the permutation argument). In [FLZ13], one considered the more general case $P_i(X) = \sigma^u$ for suitably chosen u. In those papers, it was also required that for a commitment $(A_1, A_2^{\gamma_1})$, both A_1 and $A_2^{\gamma_1}$ belong to the same group $\mathbb{G}_i$. The choice of A_1 belonging to $\mathbb{G}_1$ and $A_2^{\gamma_1}$ belonging to $\mathbb{G}_2$, as in the current paper, results in a slightly better efficiency. See [KZG10] for yet another related commitment scheme.

Gennaro et al. [GGPR13] implicitly used the $\{P_i\}$-commitment scheme, with $P_i(X)$ being dependent on the application. In the case of [GGPR13] however, one needs to prove that an (arithmetic) circuit $\mathcal{C}$ is satisfiable (or more generally that $\mathcal{C}(\boldsymbol{u}) = y$ for public y, where $\boldsymbol{u}$ is committed to), and the polynomials $P_i(X)$ depend on the concrete circuit. Thus, one only gets a non-adaptive NIZK (i.e., the CRS depends on the circuit). Adaptive soundness is achieved by using universal circuits [Val76]; in this case the polynomials $P_i(X)$ depend on the universal circuit. Presumably, because of such considerations, [GGPR13] never explicitly defined a commitment scheme.

B Product Argument: Computation of π

For the sake of simplicity, consider the case without zero knowledge, the full case is just slightly more complicated. Recall that in this case, the prover has to compute the polynomial $Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{a}\circ\boldsymbol{b}}(X) = L_{\boldsymbol{a}}(X) \cdot L_{\boldsymbol{b}}(X) - L_{\boldsymbol{a}\circ\boldsymbol{b}}(X)$. Recall that $\omega_j = \omega^{j-1}$, where ω is the nth primitive root of unity modulo p. Note that:

1. Computation of $L_{\boldsymbol{a}}(X) \cdot L_{\boldsymbol{b}}(X)$ can be performed as follows:
   (a) Use inverse FFT to compute $L_{\boldsymbol{a}}(X)$ from $\boldsymbol{a}$ and $L_{\boldsymbol{b}}(X)$ from $\boldsymbol{b}$.
   (b) Use FFT to compute $L_{\boldsymbol{a}}(\omega^j)$ for $j \in \{0, \dots, 2n-1\}$. Since $L_{\boldsymbol{a}}(\omega^j) = a_j$ for $j \in \{0, \dots, n-1\}$ is already known, some of the computation can be omitted. Denote $(\boldsymbol{a}, \boldsymbol{a'}) = (L_{\boldsymbol{a}}(\omega^j))_{j=0}^{2n-1}$.
   (c) Similarly, compute $L_{\boldsymbol{b}}(\omega^j)$, for $j \in \{n, \dots, 2n-1\}$, and $(\boldsymbol{b}, \boldsymbol{b'})$.
   (d) Compute $(\boldsymbol{a}, \boldsymbol{a'}) \circ (\boldsymbol{b}, \boldsymbol{b'})$.
   (e) Compute the polynomial $L_{\boldsymbol{a}}(X) \cdot L_{\boldsymbol{b}}(X)$ from $(\boldsymbol{a}, \boldsymbol{a'}) \circ (\boldsymbol{b}, \boldsymbol{b'})$ by using inverse FFT.
2. Computation of $L_{\boldsymbol{a}\circ\boldsymbol{b}}(X)$ can be performed as follows:
   (a) Reusing $\boldsymbol{a} \circ \boldsymbol{b}$ from a previous step, compute $L_{\boldsymbol{a}\circ\boldsymbol{b}}(X)$ by using inverse FFT.
3. Compute $Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{a}\circ\boldsymbol{b}}(X)$ by coordinate-wise subtraction.

C More on Zero Knowledge

NIZK proofs [BFM88] allow the prover to convince the verifier that an input u belongs to an **NP** language $\mathcal{L}$ in the manner that nothing else expect the truth of the statement is revealed. NIZK proofs for non-trivial languages do not exist without a trusted setup unless $\mathbf{P} = \mathbf{NP}$. There are two popular approaches to deal with this. The first approach, the use of random oracle model, results often in very efficient protocols. However, it is well known [CGH98,GK03] that some protocols that are secure in the random oracle model are non-instantiable in the standard model, and thus the random oracle model is a heuristic at its best.

A better approach is to construct NIZK proofs in the common reference string (CRS) model [BFM88]. Potentially many verifiers can then later independently verify the proof, by having access to the same CRS. In practice, one is interested in proofs where both the proof length and the verification time are sublinear in the statement size. Sublinear (adaptive) proofs can only be computationally sound, and their soundness cannot be proven under falsifiable assumptions [GW11]. (See [JR13] for a recent sublinear quasi-adaptive NIZK.) The latter means that one has to employ knowledge assumptions [Dam91]. A computationally sound proof is also known as an *argument*.

Formal Definitions. Let $\mathcal{R} = \{(u, w)\}$ be an efficiently computable binary relation with $|w| = \text{poly}(|u|)$. Here, u is a statement, and w is a witness. Let $\mathcal{L} = \{u : \exists w, (u, w) \in \mathcal{R}\}$ be an **NP**-language. Let $n = |u|$ be the input length. For fixed n, we have a relation $\mathcal{R}_n$ and a language $\mathcal{L}_n$. A *non-interactive argument* for $\mathcal{R}$ consists of three probabilistic polynomial-time algorithms: a common reference string (CRS) generator gencrs, a prover pro, and a verifier ver. For $(\mathsf{crs} = (\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(1^\kappa, n)$ (where td is not accessible to anybody but the simulator), $\mathsf{pro}(\mathsf{crs}_p; u, w)$ produces an argument π, and $\mathsf{ver}(\mathsf{crs}_v; u, \pi)$ outputs either 1 (accept) or 0 (reject).

Π is *perfectly complete*, if for all $n = \text{poly}(\kappa)$, the following probability is 1:

$$\Pr\left[\,((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(1^\kappa, n), (u, w) \leftarrow \mathcal{R}_n : \mathsf{ver}(\mathsf{crs}_v; u, \mathsf{pro}(\mathsf{crs}_p; u, w)) = 1 \quad\right] .$$

Π is adaptively *computationally sound* for $\mathcal{L}$, if for all $n = \mathrm{poly}(\kappa)$ and non-uniform probabilistic polynomial-time $\mathcal{A}$, the following probability is negligible in κ:

$$\Pr\left[\,((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(1^\kappa, n), (u, \pi) \leftarrow \mathcal{A}(\mathsf{crs}_p, \mathsf{crs}_v) : u \not\in \mathcal{L}_n \wedge \mathsf{ver}(\mathsf{crs}_v; u, \pi) = 1 \quad\right] \enspace .$$

Π is adaptively *computationally culpably sound* [GL07,GOS12] for $\mathcal{L}$, if for all $n = \mathrm{poly}(\kappa)$, for all polynomial-time decidable binary relations $\mathcal{R}^{\mathrm{guilt}} = \{\mathcal{R}_n^{\mathrm{guilt}}\}$ consisting of elements from $\bar{\mathcal{L}}$ and witnesses w^{guilt}, and for all non-uniform probabilistic polynomial-time $\mathcal{A}$, the following probability is negligible in κ:

$$\Pr\left[\begin{array}{l}((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(1^\kappa, n), (u, \pi, w^{\mathrm{guilt}}) \leftarrow \mathcal{A}(\mathsf{crs}_p, \mathsf{crs}_v) : \\ (u, w^{\mathrm{guilt}}) \in \mathcal{R}_n^{\mathrm{guilt}} \wedge \mathsf{ver}(\mathsf{crs}_v; u, \pi) = 1\end{array}\right] \enspace .$$

Π is *perfectly witness-indistinguishable*, if for all $n = \mathrm{poly}(\kappa)$, if $((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \in \mathsf{gencrs}(1^\kappa, n)$ and $((u, w_0), (u, w_1)) \in \mathcal{R}_n^2$, then the distributions $\mathsf{pro}(\mathsf{crs}_p; u, w_0)$ and $\mathsf{pro}(\mathsf{crs}_p; u, w_1)$ are equal. Π is *perfectly zero-knowledge*, if there exists a probabilistic polynomial-time simulator $\mathcal{S}$, such that for all stateful non-uniform probabilistic polynomial-time adversaries $\mathcal{A}$ and $n = \mathrm{poly}(\kappa)$,

$$\Pr\left[\begin{array}{l}((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(1^\kappa, n), \\ (u, w) \leftarrow \mathcal{A}(\mathsf{crs}_p, \mathsf{crs}_v), \\ \pi \leftarrow \mathsf{pro}(\mathsf{crs}_p; u, w) : \\ (u, w) \in \mathcal{R}_n \wedge \mathcal{A}(\pi) = 1\end{array}\right] = \Pr\left[\begin{array}{l}((\mathsf{crs}_p, \mathsf{crs}_v); \mathsf{td}) \leftarrow \mathsf{gencrs}(1^\kappa, n), \\ (u, w) \leftarrow \mathcal{A}(\mathsf{crs}_p, \mathsf{crs}_v), \\ \pi \leftarrow \mathcal{S}(\mathsf{crs}_p, \mathsf{crs}_v; u, \mathsf{td}) : \\ (u, w) \in \mathcal{R}_n \wedge \mathcal{A}(\pi) = 1\end{array}\right] \enspace .$$

Π is *a proof of knowledge*, if for all $n = \mathrm{poly}(\kappa)$ and every non-uniform probabilistic polynomial-time $\mathcal{A}$, there exists a non-uniform probabilistic polynomial-time extractor X, such that for every auxiliary input $\mathsf{aux} \in \{0,1\}^{\mathrm{poly}(\kappa)}$, the following probability is negligible in κ:

$$\Pr\left[\begin{array}{l}((\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td}) \leftarrow \mathsf{gencrs}(1^\kappa, n), ((u, \pi); w) \leftarrow (\mathcal{A} || X_{\mathcal{A}})(\mathsf{crs}_p, \mathsf{crs}_v; \mathsf{aux}) : \\ (u, w) \not\in \mathcal{R} \wedge \mathsf{ver}(\mathsf{crs}_v; u, \pi) = 1\end{array}\right] \enspace .$$

Here, the notation $\mathcal{A}||X_{\mathcal{A}}$ is defined in Sect. 2. As in the definition of PKE (see Sect. 2), we can restrict the definition of a proof of knowledge to benign auxiliary information generators, where aux is known to come from. For the sake of simplicity, we omit further discussion.

An argument that satisfies above requirements is known as *adaptive*. An argument where the CRS can depend not only on n but also on the statement u is often called *non-adaptive*. See [JR13] for a formalization of *quasi-adaptive* arguments. It is not surprising that non-adaptive (or quasi-adaptive) arguments can be much more efficient than adaptive arguments, see [GGPR13,JR13,JR14] for some examples.

D Restriction Argument

Let $\boldsymbol{e_S} = \sum_{i \in S} \boldsymbol{e_i}$. The idea is to prove that $\boldsymbol{a} \circ \boldsymbol{e_S} = \boldsymbol{a} \circ \boldsymbol{1}_n$ or equivalently, $\boldsymbol{a} \circ (\boldsymbol{e_S} - \boldsymbol{1}_n) = \boldsymbol{0}_n$. Here, the only private input is $\boldsymbol{a}$. Thus, in this case,

$$Q_{zk}^{\boldsymbol{a}, \boldsymbol{e_S} - \boldsymbol{1}_n, \boldsymbol{0}_n; r_a, 0, 0}(X) = (L_{\boldsymbol{a}}(X) + r_a Z(X))(L_{\boldsymbol{e_S}}(X) - 1) = L_{\boldsymbol{a}}(X)(L_{\boldsymbol{e_S}}(X) - 1) + r_a Z(X)(L_{\boldsymbol{e_S}}(X) - 1) \enspace .$$

Thus, $\pi_{zk}(X) = Q_{zk}(X)/Z(X) = \sum_{i \in S} a_i \ell_i(X)(L_{\boldsymbol{e_S}}(X) - 1)/Z(X) + r_a(L_{\boldsymbol{e_S}}(X) - 1)$, and

$$\pi_\times = g_1^{\pi_{zk}(\sigma)} = \prod_{i \in S} (g_1^{\ell_i(\sigma)(L_{\boldsymbol{e_S}}(\sigma)-1)/Z(\sigma)})^{a_i} \cdot (g_1^{L_{\boldsymbol{e_S}}(\sigma)-1})^{r_a} \enspace .$$

Assuming S is known in advance (which is often the case), $g_1^{\ell_i(\sigma)(L_{\boldsymbol{e_S}}(\sigma)-1)/Z(\sigma)}$ and $g_1^{L_{\boldsymbol{e_S}}(\sigma)-1}$ can be put to the CRS. Then, one can compute $\pi_\times$ by using $|S| + 1$ exponentiations. The argument will consist of a single group element, while the verifier has to execute two pairings to check that for $(T_1, T_2^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{e_S}; 0)$, $\hat{e}(A_1, T_2^{\gamma_1}/g_2^{\gamma_1}) = \hat{e}(\pi_\times, g_2^{\gamma_1 Z(\sigma)})$. Also $T_2^{\gamma_1}/g_2^{\gamma_1}$ can be stored in the CRS.

The new restriction argument has the same complexity as the restriction argument of [Gro10]. However, as in the case of the general product argument, it does not use an extra knowledge assumption to derive culpable soundness of the argument but instead relies on a purely computational assumption (TSDH).

E Proof of Thm. 3

Proof. First, assuming that $(B_1, B_2^{\gamma_2})$ is a correct commitment, $\hat{e}(g_1^{Z(\sigma)}, B_2^{\gamma_2}\pi_2^{\gamma_2}) = \hat{e}(g_1^{Z(\sigma)}, B_2^{\gamma_2})\hat{e}(g_1^{Z(\sigma)}, \pi_2^{\gamma_2}) = \hat{e}(B_1, g_2^{\gamma_2 Z(\sigma)})\hat{e}(g_1^{Z(\sigma)}, \pi_2^{\gamma_2})$ and thus the first part of verification equation holds if and only if $\hat{e}(g_1, \pi_2^{\gamma_2}) = \hat{e}(\pi_1, g_2^{\gamma_2})$. By writing the verification like that, we saved one pairing. (Moreover, we added extra $Z(\sigma)$ to exponents so that the verification equation be computable from $\mathsf{crs}_{\mathsf{rsft}}$.)

Completeness: follows from the derivation of the argument. Witness-indistinguishability: since argument π_{rsft} that satisfies the verification equations is unique, all witnesses result in the same argument, and hence the product argument is witness-indistinguishable.

Culpable soundness: Assume that the PKE assumption holds. First, we recall that in this case the culpable soundness means that a non-uniform probabilistic polynomial time adversary against the shift argument of the current section has negligible chance, given $\mathsf{crs}_{\mathsf{rsft}} \leftarrow \mathsf{gencrs}(1^\kappa, n)$ as an input, of outputting common input $\mathsf{inp}_{\mathsf{rsft}} = (A_1, A_2^{\gamma_1}, B_1, B_2^{\gamma_1})$, an accepting argument $\pi_{\mathsf{rsft}} \leftarrow (\pi_1, \pi_2^{\gamma_2})$, and a witness $(\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, (\pi_f^*)_{f \in \Phi_{z-\mathsf{rsft}}})$ with $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}_p^n$, $r_a, r_b \in \mathbb{Z}_p$, and $\pi_f^* \in \mathbb{Z}_p$ for $f \in \Phi_{z-\mathsf{rsft}}$, such that

(i) $\mathsf{ver}_{\mathsf{rsft}}(\mathsf{crs}_v; \mathsf{inp}_{\mathsf{rsft}}; w_{\mathsf{rsft}})$ accepts,
(ii) $(\pi_1, \pi_2^{\gamma_2}) = (g_1, g_2^{\gamma_2})^{\pi^*(\sigma)}$ where $\pi^*(X) = \sum_{f \in \Phi_{z-\mathsf{rsft}}} \pi_f^* \cdot f(X)$ (this follows from the PKE assumption and (i)),
(iii) $(A_1, A_2^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r_a)$, $(B_1, B_2^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{b}; r_b)$, and
(iv) $(a_n, \dots, a_1) \neq (0, \dots, 0, b_n, \dots, b_{z+1})$.

Assume that $\mathcal{A}_{sound}$ is an adversary that outputs the common input, an accepting argument, the witness, and the coefficients π_f^* of $\pi(X)$ such that (i–iv) hold. We now construct the following adversary $\mathcal{A}_{pdl}$ that breaks PDL. It knows all coefficients in Eq. (10), and thus has obtained coefficients of a polynomial $d(X)$, such that $d(\sigma) = 0$. If the prover is dishonest, then $d(X)$ is a non-zero polynomial. In this case, $\mathcal{A}_{pdl}$ can use an efficient polynomial factorization algorithm [LLL82,vHN10] to find all roots r_i of $d(X)$. One of those roots has to be σ; this can be tested by comparing say the values $g_1^{Z(r_i)Z(\sigma)}$ with the value $g_1^{Z(\sigma)^2}$ given in the CRS. (Again, inclusion of an extra $Z(\sigma)$ to exponents means that we do not have to require that we can perform verification without adding more elements to the CRS.) □

F Proof of Lem. 2

Proof. By Remark 1, we must show that $\mathcal{F}_{1-\mathsf{rsft},1}$ consists of $2n+1$ linearly independent polynomials of degree $\leq 2n$. Assume that $\sum_{i=2}^{n} a_i(\ell_{i-1}(X)Z(X) - \ell_i(X)) + \sum_{i=1}^{n} b_i \ell_i(X) + cZ(X)^2 + dZ(X) = 0$. Then, similarly to the proof of Lem. 1, $-\sum_{i=2}^{n} a_i \boldsymbol{e_i} + \boldsymbol{b} = \mathbf{0}_n$. That is, $b_1 = 0$ and $b_i = a_i$ for $i > 1$. Thus, $\sum_{i=2}^{n} b_i \ell_{i-1}(X) + cZ(X) + d = 0$. Analogously, this means that $\boldsymbol{b} + d\mathbf{1}_n = \mathbf{0}_n$. Since $b_1 = 0$, this is only possible when $b_1 = \cdots = b_n = 0$ and $d = 0$. Thus, $c = 0$. Hence, $\mathcal{F}_{1-\mathsf{rsft},1}$ is linearly independent. The rest is straightforward. □

G Proof of Thm. 4

Proof. Completeness: $\boldsymbol{S} \in$ Subset-Sum iff there exists $\boldsymbol{b} \in \{0,1\}^n$ such that $\sum_{i=1}^{n} S_i b_i = s$. Here, π_1 proves that $b_i \in \{0,1\}$, π_2 proves that $c_i = S_i b_i$, $(\pi_{31}, \pi_{32}^{\gamma_2})$ proves that $d_j - c_j = d_{j+1}$ for $j < n$ and $d_n - c_n = 0$ (and thus $d_n = c_n$, $d_{n-1} = c_{n-1} + d_n$ and in general $d_j = \sum_{i=j}^{n} c_i = \sum_{i=j}^{n-1} S_i b_i$), and finally π_4 proves that $d_1 = s$. Thus, $\sum_{i=1}^{n} S_i b_i = s$ and therefore, $\sum_{i=1}^{n} S_i b_i = s$.

Adaptive computational soundness follows, under the corresponding assumptions on genbp, from the culpable soundness of every basic argument. First, by Lem. 2, $(\mathcal{F}^*_{1-\mathsf{rsft},1}, \mathcal{F}^*_{1-\mathsf{rsft},2})$-PDL is equal to the $(\mathcal{F}_{1-\mathsf{rsft},1}, \mathcal{F}_{1-\mathsf{rsft},2})$-PDL assumption. Moreover, by a remark in Sect. 2, $(\mathcal{F}^*_{\times,1}, \mathcal{F}^*_{\times,2}, \{\omega_i\}_{i=1}^{n}, 1)$-TSDH is equal to the $(\mathcal{F}_{\times,1}, \mathcal{F}_{\times,2}, \{\omega_i\}_{i=1}^{n}, 1)$-TSDH assumption and $(\mathcal{F}^*_{\mathsf{com},1}, \dots)$-PKE is equal to the $(\mathcal{F}_{\mathsf{com},1}, \dots)$-PKE assumption.

Assume that both PKE assumptions hold and that there exists an adversary $\mathcal{A} = \mathcal{A}_{sound}$ that breaks the soundness of the Subset-Sum argument. We construct an adversary $\mathcal{A}_{dl}$ (resp., $\mathcal{A}_{tsdh}$) that breaks the corresponding PDL (resp., TSDH) assumption on genbp as follows. First,

(i) since genbp is $(\mathcal{F}^*_{\mathsf{com},1}, \mathcal{F}_{\mathsf{ssum},1} \setminus \mathcal{F}_{\mathsf{com},1}, Y_2 \mathit{\Phi}_{1-\mathsf{rsft}}, 1)$-PKE secure, there exists an extractor that obtains $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$ (and the used randomizers r_b, r_c and r_d) from $(B_1, B_2^{\gamma_1})$, $(C_1, C_2^{\gamma_1})$, and $(D_1, D_2^{\gamma_1})$.

(ii) From π_1: due to (i), $\mathcal{A}_{tsdh}$ has access to $\boldsymbol{b}$, and hence by the TSDH assumption the culpable soundness of the product argument, $b_i \in \{0,1\}$.

(iii) From π_2: due to (i), $\mathcal{A}_{tsdh}$ has access to $\boldsymbol{b}$ and $\boldsymbol{c}$, and hence by the TSDH assumption and the culpable soundness of the product argument, $\boldsymbol{c} = \boldsymbol{S} \circ \boldsymbol{b}$.

(iv) From $(\pi_{31}, \pi_{32}^{\gamma_2})$: since genbp is $(\mathit{\Phi}_{1-\mathsf{rsft}}, \mathcal{F}_{\times,2}, \mathcal{F}_{\times,2}, 2)$-PKE secure, there exists an extractor that obtains a witness $(\pi^*_f)_f$ that the argument belongs to the correct span. Due to this, $\mathcal{A}_{dl}$ has access to all values required in Thm. 3, and hence by the $(\mathcal{F}^*_{1-\mathsf{rsft},1}, \mathcal{F}^*_{1-\mathsf{rsft},2})$-PDL assumption and the culpable soundness of the shift argument, $d_i = \sum_{j \geq i} c_j = \sum_{j \geq i} S_j b_j$ for all i.

(v) From π_4: due to (i), $\mathcal{A}_{tsdh}$ has access to $\boldsymbol{d}$, and hence by the TSDH assumption and the culpable soundness of the product argument, $\boldsymbol{e_1} \circ (\boldsymbol{d} - s\boldsymbol{e_1}) = \mathbf{0}_n$ and thus $d_1 = \sum_{i=1}^{n} S_i b_i = s$.

Argument of knowledge: follows the above proof of soundness, since there exists an extractor that retrieves the witness $(\boldsymbol{b}, r_b, \boldsymbol{c}, r_c, \boldsymbol{d}, r_d)$ that corresponds to the satisfying argument.

Perfect zero-knowledge: follows from the presence of the trapdoor (this allows the simulator to open all commitments to $\mathbf{0}_n$), and from the facts that $\mathbf{0}_n \circ \mathbf{0}_n = \mathbf{0}_n$ and that $\mathbf{0}_n$ is a shift of $\mathbf{0}_n$. The simulator creates $(B_1, B_2^{\gamma_1})$, $(C_1, C_2^{\gamma_1})$ and $(D_1, D_2^{\gamma_1})$ as random commitments to $\mathbf{0}_n$, and uses the knowledge of td (and the trapdoor property of the commitment scheme) to compute r_e^* such that $\mathsf{com}(\mathsf{ck}; \boldsymbol{e_1}; \mathbf{0}) = \mathsf{com}(\mathsf{ck}; \mathbf{0}_n; r_e^*)$. She then simulates all four basic arguments, based on her knowledge of the trapdoor. All product arguments are obviously correct when the committed values are equal to $\mathbf{0}_n$: $\mathbf{0}_n = \mathbf{0}_n \circ \mathbf{0}_n$ (this takes care of π_1), $\mathbf{0}_n = \boldsymbol{S} \circ \mathbf{0}_n$ (this takes care of π_2), $\mathbf{0}_n$ is a right shift of $\mathbf{0}_n$ (this takes care of $(\pi_{31}, \pi_{32}^{\gamma_2})$), and $\mathbf{0}_n \circ (\mathbf{0}_n - s\mathbf{0}_n) = \mathbf{0}_n$ (this takes care of π_4).

Finally, witness-indistinguishability of the basic arguments guarantees that the simulated argument comes from the same distribution as the real argument. This finishes the proof. □

H On Using A Presumably Weaker Knowledge Assumptions

We will now describe a variant of the shift argument from Sect. 6 that makes it possible for the Subset-Sum argument of Sect. 7 to rely on a presumably weaker version of the PKE assumption. Namely, instead of a $(\varPhi_{1-\mathsf{rsft}}, \dots)$-PKE assumption, the range argument will use an $(\{X^i\}_{i\in[0\,..\,d]}, \dots)$-PKE assumption for some integer $d = \Theta(n)$. This weakening is based on the ideas of [GGPR13]. As a trade-off, the computational assumption will become presumably stronger (a CDH-type assumption instead of a DL-type assumption). Moreover, the modified shift argument will be somewhat less efficient. We emphasize that such a weakening of the PKE assumption is important in [GGPR13] where without it, one would have to rely on an $(\mathcal{F}, \dots)$-PKE assumption where $\mathcal{F}$ depends on the statement to be proven. The latter is not true in our case , where $\varPhi_{1-\mathsf{rsft}}$ does not depend on the concrete instance of Subset-Sum.

Modified Shift Argument. The small number of modifications, compared to the shift argument of Sect. 6, follow the same methodology as the main security proof of [GGPR13] so we will omit a longer explanation. The description of the modified shift argument follows:

CRS generation $\mathsf{gencrs}_{\mathsf{rsft}}(1^\kappa, n)$**:** Let $\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, \hat{e}) \leftarrow \mathsf{genbp}(1^\kappa)$, and pick $g_2 \leftarrow_r \mathbb{G}_2^*$. Generate $(\sigma, \alpha, \beta, \gamma_1, \gamma_2) \leftarrow \mathbb{Z}_p^5$ with $Z(\sigma) \neq 0$ and $\gamma_1 \neq 0$. Set $\mathsf{ck} \leftarrow (\mathsf{gk}; (g_1, g_2^{\gamma_1})^{\mathcal{F}_{\mathsf{com},1}(\sigma)})$. Let $d := \max(2n, n+3)$. Let $\mathsf{crs}_p \leftarrow (\mathsf{gk}; \mathsf{ck}, ((g_1, g_2^{\gamma_2})^{\sigma^i})_{i\in[0\,..\,d]}, g_1^{\beta\varPhi_{z-\mathsf{rsft}}(\sigma)})$, $\mathsf{crs}_v \leftarrow (\mathsf{gk}; g_1, (g_1, g_2^{\gamma_2})^{Z(\sigma)}, (g_2, g_2^{\beta})^{\alpha})$. Set $\mathsf{td} \leftarrow (\sigma, \alpha, \beta, \gamma_1, \gamma_2)$. Return $(\mathsf{crs} = (\mathsf{crs}_p, \mathsf{crs}_v), \mathsf{td})$.
Common inputs: $\mathsf{inp}_{\mathsf{rsft}} = (A_1, A_2^{\gamma_1}, B_1, B_2^{\gamma_1})$.
Proving $\mathsf{pro}_{\mathsf{rsft}}(\mathsf{crs}_p; \mathsf{inp}_{\mathsf{rsft}}; \boldsymbol{a}, r_a, \boldsymbol{b}, r_b)$**:** compute $\pi_{\mathsf{rsft}} \leftarrow (g_1, g_1^{\beta}, g_2^{\gamma_2})^{\pi(\sigma)} = (\pi_1, \pi_1^{\beta}, \pi_2^{\gamma_2})$ analogously to Eq. (10). Return π_{rsft}.
Verification $\mathsf{ver}_{\mathsf{rsft}}(\mathsf{crs}_v; \mathsf{inp}_{\mathsf{rsft}}; \pi_{\mathsf{rsft}} = (\pi_1, \pi_1^{\beta}, \pi_2^{\gamma_2}))$**:** check that
 (i) $\hat{e}(\pi_1, g_2^{\gamma_2}) = \hat{e}(g_1, \pi_2^{\gamma_2})$,
 (ii) $\hat{e}(g_1^{Z(\sigma)}, B_2^{\gamma_2}\pi_2^{\gamma_2}) = \hat{e}(B_1\pi_1, g_2^{\gamma_2 Z(\sigma)}) = \hat{e}(A_1, g_2^{\gamma_2 Z(\sigma)})$, and
 (iii) $\hat{e}(\pi_1^{\beta}, g_2^{\alpha}) = \hat{e}(\pi_1, g_2^{\beta\alpha})$.

To prove culpable soundness, we will use the following computational assumption, variants of which are well-known [GJM02,BBG05,Gro10,GGPR13]. Assume that $1 < d < d^* = \mathrm{poly}(\kappa)$ are two integers. Then, genbp is (d, d^*)*-PCDH (Power Computational Diffie-Hellman) secure* if for any non-uniform probabilistic polynomial-time adversary $\mathcal{A}$, the following probability is negligible in κ:

$$\Pr\left[\mathsf{gk} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, \hat{e}) \leftarrow \mathsf{genbp}(1^\kappa, n), \sigma \leftarrow \mathbb{Z}_p : \mathcal{A}\left(\mathsf{gk}; (g_1^{\sigma^i})_{i\in[0\,..\,d^*]\setminus\{d+1\}}\right) = g_1^{\sigma^{d+1}}\right] .$$

The PCDH assumption is also a variant of the uber-assumption [BBG05]. Being a power variant of the Computational Diffie-Hellman assumption, it is (presumably) stronger than the PDL assumption that is a power variant of the discrete logarithm assumption.

Theorem 5. *Let* $n = \mathrm{poly}(\kappa)$*. Let* com *be the interpolating commitment scheme. The shift argument of the current section is perfectly complete and perfectly witness-indistinguishable. Let* $\varPhi_{z-\mathsf{rsft}}$ *be as in Lem. 1. Let* $d = \max(2n, n+3)$ *and* $d^* = 2n + d$*. If* genbp *is* (d, d^*)*-PCDH and* $(\{X^i\}_{i\in[0\,..\,d]}, X_\beta\varPhi_{z-\mathsf{rsft}}, \{X_\alpha, X_\alpha X_\beta\}, 2)$*-PKE secure, then this argument is adaptively culpably sound.*

Proof. Completeness: follows from the derivation of the argument. Witness-indistinguishability: since argument π_{rsft} that satisfies the verification equations is unique, all witnesses result in the same argument, and hence the product argument is witness-indistinguishable.

Culpable soundness: In this case, culpable soundness together with the PKE assumption means that a non-uniform probabilistic polynomial time adversary against the shift argument of the current section has negligible chance, given correctly generated crs as an input, of outputting common input $\mathsf{inp}_{\mathsf{rsft}} = (A_1, A_2^{\gamma_1}, B_1, B_2^{\gamma_1})$ and an accepting argument $\pi_{\mathsf{rsft}} \leftarrow (\pi_1, \pi_1^{\beta}, \pi_2^{\gamma_2})$ together with a witness $(\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, (\pi_i^*)_{i=0}^d)$ with $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}_p^n$, $r_a, r_b \in \mathbb{Z}_p$ and $\pi^*(X) = \sum_{i=0}^d \pi_i^* X^i \in \mathbb{Z}_p[X]$, such that

(i) $(\pi_1, \pi_2^{\gamma_2}) = (g_1, g_2^{\gamma_2})^{\pi^*(\sigma)}$,
(ii) $(A_1, A_2^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r_a)$, $(B_1, B_2^{\gamma_1}) = \mathsf{com}(\mathsf{ck}; \boldsymbol{b}; r_b)$, and
(iii) $(a_n, \ldots, a_1) \neq (0, \ldots, 0, b_n, \ldots, b_{z+1})$.

Let $\mathcal{A}_{sound}$ be an adversary that with some non-negligible probability ε replies with an accepting argument such that conditions (i–iii) hold. By previous discussion this means that $\pi^*(X)$ is not in the span of $\Phi_{z-\mathsf{rsft}}$. Recall that $\Phi_{z-\mathsf{rsft}}$ consists of degree-$\leq 2n$ polynomials. We construct the following d-PCDH adversary $\mathcal{A}_{pcdh}$. She receives a d-PCDH challenge $ch_{pcdh} = (\mathsf{gk}; (g_1^{\sigma^i})_{i \in [0\,..\,d^*] \setminus \{d+1\}})$ for random σ. Let

$$D := \{a(X) \in \mathbb{Z}_p[X] : \deg a(X) \leq d \wedge a(X)f(X) \text{ has a zero coefficient for } X^d \text{ for all } f(X) \in \Phi_{z-\mathsf{rsft}}\} \enspace .$$

She picks $a(X) \leftarrow_r D$ randomly. Note that $\deg(a(X)f(X)) \leq d + 2n$ for any $f(X) \in \Phi_{z-\mathsf{rsft}}$. There are $(d+1) - |\Phi_{z-\mathsf{rsft}}| \geq (n+3) - (n+2) = 1 > 0$ degrees of freedom for choosing $a(X)$. Therefore, for a polynomial $\pi^*(X)$ outside of the span of $\Phi_{z-\mathsf{rsft}}$, the coefficient of X^d of $a(X)\pi^*(X)$ will be random.

Next, $\mathcal{A}_{pcdh}$ picks $b \leftarrow_r \mathbb{Z}_p$, and sets $\beta(X) \leftarrow a(X)X + b$ and $\beta \leftarrow \beta(\sigma)$. Since $a(X) \in D$ and $\deg f \leq 2n$ for $f \in \Phi_{z-\mathsf{rsft}}$, she can compute $g_1^{\beta f(\sigma)} = g_1^{(a(\sigma)\sigma + b)f(\sigma)}$ from ch_{pcdh}. Now, she picks $(\alpha, \gamma_1, \gamma_2) \leftarrow_r \mathbb{Z}_p^3$. She computes correct CRS based on already known values, and sends it to $\mathcal{A}_{sound}$. Assume that $\mathcal{A}_{sound}$ answers with $(\boldsymbol{a}, r_a, \boldsymbol{b}, r_b; (\pi_i^*)_{i \in [0\,..\,d]}; \pi_1, \pi_1^{\beta}, \pi_2^{\gamma_2})$ such that the verification succeeds and $\pi_1 = g_1^{\pi^*(\sigma)}$, where $\pi^*(X) = \sum_{i=0}^d \pi_i^* X^i$. Since b is random, β does not reveal any information about $a(X)$. Hence, the coefficient of X^d in $a(X)\pi^*(X)$ is random. Thus, with probability $1 - 1/p$, the coefficient of X^{d+1} in $\beta(X)\pi^*(X)$ (a polynomial of degree $\leq 2d+1 \leq 2n+d$) is non-zero. In this case, since $\mathcal{A}_{pcdh}$ knows the coefficients of the polynomial $\beta(X)\pi^*(X)$, she can compute $g_1^{\sigma^{d+1}}$ from $\pi_1^{\beta} = g_1^{\beta\pi^*(\sigma)}$ and ch_{pcdh}. Thus, $\mathcal{A}_{pcdh}$ can break the (d, d^*)-PCDH assumption with probability $(1 - 1/p) \cdot \varepsilon$. □

This version of the shift argument is somewhat less efficient than the one in Sect. 6. The prover computation is dominated by three (instead of two) $(n+2)$-wide multi-exponentiations. The communication is three (instead of two) group elements. The verifier computation is dominated by 7 (instead of 5) pairings. Apart from gk, and assuming that $d = 2n$, the prover CRS contains $2((n+1)+d) + (n+2) = 7n+4$ group elements, and the verifier CRS contains 5 group elements.

Modified Subset-Sum Argument. We can clearly just use this version of the shift argument in the new Subset-Sum argument of Sect. 7. The Subset-Sum argument will remain secure but will rely on a different set of assumptions. Note that apart from using the PCDH assumption and a weaker PKE assumption, the assumptions of Sect. 2 have to be slightly modified to allow for a longer secret key. (Namely, everywhere σ has to be replaced with (σ, α, β) and thus polynomials $f(X, \boldsymbol{Y})$ have to be replaced with 3-variate polynomials $f(X, X_\alpha, X_\beta, \boldsymbol{Y})$, where X_α corresponds to α and X_β corresponds to β.) We also redefine the polynomial sets as $\hat{\mathcal{F}}^1_{\mathsf{rsft},1} := \mathcal{F}_{\mathsf{com},1} \cup X_\beta \varPhi_{z-\mathsf{rsft}}$ and $\hat{\mathcal{F}}^2_{\mathsf{rsft},1} := \mathcal{F}_{\mathsf{com},2} \cup \{X_\alpha, X_\alpha X_\beta\}$, and define $\hat{\mathcal{F}}_{\mathsf{ssum},1}$ and $\hat{\mathcal{F}}_{\mathsf{ssum},2}$ as in Sect. 7 but by using the redefined sets $\hat{\mathcal{F}}^1_{\mathsf{rsft},1}$ and $\hat{\mathcal{F}}^2_{\mathsf{rsft},1}$ from the current paragraph.

Theorem 6. *Let* $n = \mathrm{poly}(\kappa)$*, and let* com *be the interpolating commitment scheme. Assume that we use the shift argument of the current section. The new* Subset-Sum *argument is perfectly complete and perfectly zero-knowledge. It is (adaptively) computationally sound and an argument of knowledge if* genbp *is* $(\mathcal{F}^*_{\times,1}, \mathcal{F}^*_{\times,2}, \{\omega_i\}^n_{i=1}, 1)$*-TSDH,* (d, d^*)*-PCDH,* $(\mathcal{F}^*_{\mathsf{com},1}, \mathcal{F}_{\mathsf{ssum},1} \setminus \mathcal{F}_{\mathsf{com},1}, Y_2 \varPhi_{1-\mathsf{rsft}}, 1)$*-PKE, and* $(\{X^i\}_{i\in[0\,..\,d]}, \mathcal{F}_{\times,1} \cup X_\beta \varPhi_{z-\mathsf{rsft}}, \mathcal{F}_{\times,2} \cup \{X_\alpha, X_\alpha X_\beta\}, 2)$*-PKE secure.*

Clearly, the modified Subset-Sum argument will also be less efficient than the Subset-Sum argument of Sect. 7. More precisely, it increases the communication by 1 group element, the prover computation by a factor of 1.5, and the verifier computation by 2 pairings.

I New Range Argument

In a range argument, given public range $[L\,..\,H]$, the prover aims to convince the verifier that he knows how to open commitment $(A_1, A_2^{\gamma_1})$ to a value $a \in [L\,..\,H]$.

We first remark that instead of the range $[L\,..\,H]$, one can consider the range $[0\,..\,H-L]$, and then use the homomorphic properties of the commitment scheme to add L to the committed value. Therefore, we will just assume that the range is equal to $[0\,..\,H]$ for some $H \geq 1$.

Assume that the common input $(A_1, A_2^{\gamma_1})$ is a commitment to vector $\boldsymbol{a}$ with $a_1 = a$ and $a_i = 0$ for $i > 1$. Let $n = \lfloor \log_2 H \rfloor + 1$. To prove that $a \in [0\,..\,H]$, we do the following.

The CRS generation $\mathsf{gencrs}_{\mathsf{ssum}}$ invokes the CRS generations of the commitment scheme, the product argument and the shift argument, sharing the same gk and trapdoor $\mathsf{td} = (\sigma, \gamma_1, \gamma_2)$ between the different invocations. The CRS also contains a commitment to $\boldsymbol{H}$ (defined in the argument), needed for a later efficient verification of the argument π_2.

The prover does the following (further explanations are given in the completeness proof):

```
Let $a = \sum_{i=1}^n H_i b_i$ for $H_i = \lfloor (H + 2^{i-1})/2^i \rfloor$ and $b_i \in \{0, 1\}$.
Let $(B_1, B_2^{\gamma_1})$ be a commitment to $\boldsymbol{b}$.
Construct a product argument $\pi_1$ to show that $\boldsymbol{b} = \boldsymbol{b} \circ \boldsymbol{b}$.
Let $(C_1, C_2^{\gamma_1})$ be a commitment to $\boldsymbol{c} \leftarrow \boldsymbol{H} \circ \boldsymbol{b}$.
Construct a product argument $\pi_2$ to show that $\boldsymbol{c} = \boldsymbol{H} \circ \boldsymbol{b}$.
Let $(D_1, D_2^{\gamma_1})$ be a commitment to $\boldsymbol{d}$, where $d_i = \sum_{j \geq i} c_i$.
Construct a shift argument $(\pi_{31}, \pi_{32}^{\gamma_2})$ to show that $\boldsymbol{d} - \boldsymbol{c}$ is a right shift-by-1 of $\boldsymbol{d}$.
Construct a product argument $\pi_4$ to show that $\boldsymbol{e_1} \circ (\boldsymbol{d} - \boldsymbol{a}) = \boldsymbol{0}_n$.
Output $\pi_{\mathsf{ssum}} = (B_1, B_2^{\gamma_1}, C_1, C_2^{\gamma_1}, D_1, D_2^{\gamma_1}, \pi_1, \pi_2, \pi_{31}, \pi_{32}^{\gamma_2}, \pi_4)$.
```

After receiving π_{ssum}, the verifier checks the validity of four commitments $(A_1, A_2^{\gamma_1})$, $(B_1, B_2^{\gamma_1})$, $(C_1, C_2^{\gamma_1})$ (by using a commitment to $\boldsymbol{H}$ that is given in the CRS), and $(D_1, D_2^{\gamma_1})$, and then verifies four basic arguments.

Let $\mathcal{F}_{\mathsf{ssum},1} = \mathcal{F}_{\times,1} \cup \varPhi_{1-\mathsf{rsft}}$ and $\mathcal{F}_{\mathsf{ssum},2} = \mathcal{F}_{\times,2} \cup Y_2 \varPhi_{1-\mathsf{rsft}}$. Clearly, here it suffices to take $\mathsf{crs} = (\mathsf{gk}; g_1^{\mathcal{F}_{\mathsf{ssum},1}(\sigma)}, g_2^{\gamma_1 \mathcal{F}_{\mathsf{com},1}(\sigma)}, g_2^{\gamma_2 \varPhi_{1-\mathsf{rsft}}(\sigma)}) = (\mathsf{gk}; g_1^{\mathcal{F}_{\mathsf{ssum},1}(\sigma)}, g_2^{\mathcal{F}_{\mathsf{ssum},2}(\sigma, \gamma_1, \gamma_2)})$.

Theorem 7. *Let* $n = \mathrm{poly}(\kappa)$*, and let* com *be the interpolating commitment scheme. The new range argument is perfectly complete and perfectly zero-knowledge. It is (adaptively) computationally sound and an argument of knowledge if* genbp *is* $(\mathcal{F}^*_{\times,1} = \{X^i\}_{i=0}^n, \mathcal{F}^*_{\times,2} = \{Y_1 X^i\}_{i=0}^n, \{\omega_i\}_{i=1}^n, 1)$*-TSDH,* $(\mathcal{F}^*_{1-\mathsf{rsft},1} = \{X^i\}_{i=0}^{2n}, \mathcal{F}^*_{1-\mathsf{rsft},2} = \{Y_2 X^i\}_{i=0}^{2n})$*-PDL,* $(\mathcal{F}^*_{\mathsf{com},1} = \{X^i\}_{i=0}^n, \mathcal{F}_{\mathsf{ssum},1} \setminus \mathcal{F}_{\mathsf{com},1}, Y_2 \Phi_{1-\mathsf{rsft}}, 1)$*-PKE, and* $(\Phi_{1-\mathsf{rsft}}, \mathcal{F}_{\times,1}, \mathcal{F}_{\times,2}, 2)$*-PKE secure.*

Proof (Sketch). Completeness: $a \in [0 \,..\, H]$ iff $a = \sum_{i=1}^n H_i b_i$ for some $b_i \in \{0,1\}$ [LAN02] (see [CLs10] for a formal proof). Here, π_1 proves that b_i are Boolean, π_2 proves that $c_i = H_i b_i$, $(\pi_{31}, \pi_{32}^{\gamma_2})$ proves that $d_j - c_j = d_{j+1}$ for $j < n$ and $d_n - c_n = 0$ (and thus $d_n = c_n$, $d_{n-1} = c_{n-1} + d_n$ and in general $d_j = \sum_{i=j}^n c_i = \sum_{i=j}^{n-1} H_i b_i$), and finally π_4 proves that $\boldsymbol{a} = (0, \dots, 0, a)$ with $d_1 - a = \sum_{i=1}^n H_i b_i - a = 0$. Thus, $a = \sum_{i=1}^n H_i b_i$ and therefore, $a \in [0 \,..\, H]$.

Perfect zero-knowledge: follows from the presence of the trapdoor (this allows the simulator to open all commitments to $\mathbf{0}_n$), and from the facts that $\mathbf{0}_n \circ \mathbf{0}_n = \mathbf{0}_n$ and that $\mathbf{0}_n$ is a shift of $\mathbf{0}_n$. The simulator creates $(B_1, B_2^{\gamma_1})$ and $(C_1, C_2^{\gamma_1})$ as random commitments to $\mathbf{0}_n$, computes $(D_1, D_2^{\gamma_1}) \leftarrow (A_1, A_2^{\gamma_1}) \cdot \mathsf{com}(\mathsf{ck}; \mathbf{0}_n; r_d^*)$ for a random r_d^*, and uses the knowledge of σ (and the trapdoor property of the commitment scheme) to compute r_e^* such that $\mathsf{com}(\mathsf{ck}; \boldsymbol{e_1}; \mathbf{0}) = \mathsf{com}(\mathsf{ck}; \mathbf{0}_n; r_e^*)$. She then simulates the basic arguments, based on her knowledge of the trapdoor. All product arguments are obviously correct when the committed values are equal to $\mathbf{0}_n$: $\mathbf{0}_n = \mathbf{0}_n \circ \mathbf{0}_n$ (this takes care of π_1), $\mathbf{0}_n = \boldsymbol{H} \circ \mathbf{0}_n$ (this takes care of π_2), $\mathbf{0}_n$ is a right shift of $\mathbf{0}_n$, and $\mathbf{0}_n \circ \mathbf{0}_n = \mathbf{0}_n$ (this takes care of π_4).

To simulate the shift argument, the simulator sets $(\pi_{31}, \pi_{32}^{\gamma_2}) \leftarrow (D_1, (D_2^{\gamma_1})^{\gamma_2/\gamma_1})^{Z(\sigma)-1} \cdot (C_1, (C_2^{\gamma_1})^{\gamma_2/\gamma_1})$. (Recall that $\gamma_1 \neq 0$.) Clearly, the first verification succeeds. For the second verification, note that $\hat{e}((D_1/C_1) \cdot \pi_{31}, g_2^{\gamma_2 Z(\sigma)}) = \hat{e}(D_1^{Z(\sigma)}, g_2^{\gamma_2 Z(\sigma)}) = \hat{e}(D_1, g_2^{\gamma_2 Z(\sigma)^2})$. Thus $(\pi_{31}, \pi_{32}^{\gamma_2})$ is an accepting shift argument.

Finally, witness-indistinguishability of the basic arguments guarantees that the simulated argument comes from the same distribution as the real argument. This finishes the proof.

(Adaptive) computational soundness: This part of the proof is very similar to the proof of Thm. 4 and thus omitted. □

Efficiency of Range Argument. The prover computation is dominated by three commitments and the application of three product arguments and one shift argument, that is, by $\Theta(n \log n)$ non-cryptographic operations and $\Theta(n)$ cryptographic operations. The latter is dominated by 11 ($\approx n$)-wide multi-exponentiations (2 in each commitment and shift argument, and 1 in each product argument). The argument size is constant (11 group elements), and the verifier computation is dominated by 20 pairings (3 pairings to verify π_2, 2 pairings to verify each of the other product arguments, 5 pairings to verify the shift argument, and 8 pairings to verify the validity of 4 commitments). In this case, since the verifier does not have to commit to $\boldsymbol{H}$, the verifier computation is dominated by $\Theta(1)$ cryptographic operations.

The resulting range argument is hence significantly more computation-efficient for the prover than the previous arguments [CLZ12,FLZ13]. For example, the latter had prover computation $\Theta(r_3^{-1}(n) \log n)$. It has better communication (11 versus 31 group elements in [FLZ13]), and verification complexity (20 versus 65 pairings in [FLZ13]). Moreover, it is also simpler: since the prover computation is quasi-linear, we do not have to consider various trade-offs (though they are still available) between computation and communication as in [CLZ12,FLZ13].

J Other Suitable Numerical NP-Complete languages

In this section, we give a brief description of some other NP-complete languages for which one can construct efficient adaptive short NIZK arguments by using the product-and-shift framework. We define the languages by using notation that facilitates design of such arguments.

PARTITION *[GJ79, p. 223].* PARTITION is the set of all vectors $\boldsymbol{S} = (S_1, \ldots, S_n)$, such that there exist $\boldsymbol{b} \in \{0,1\}^n$ with $\sum_{i=1}^n S_i b_i = \frac{1}{2} \cdot \sum_{i=1}^n S_i$. Thus, PARTITION is a special case of SUBSET-SUM with $s = \frac{1}{2} \cdot \sum_{i=1}^n S_i$. A PARTITION argument follows trivially from the SUBSET-SUM arguments.

SUBSET-PRODUCT *[GJ79, p. 224].* SUBSET-PRODUCT is the set of all vectors $(\boldsymbol{S} = (S_1, \ldots, S_n), s)$, such that there exist $\boldsymbol{b} \in \{0,1\}^n$ with $\prod_{i:b_i=1} S_i b_i = s$. The resulting argument is almost the same as the SUBSET-SUM argument, except one step. Namely, the prover computes still a vector $\boldsymbol{c} = \boldsymbol{S} \circ \boldsymbol{b}$. However, differently from the SUBSET-SUM argument of Sect. 7, the prover now lets $\boldsymbol{d}$ to be the multiplicative scan of $\boldsymbol{c}$ with $d_i = \prod_{j \geq i} c_j$, and then proves the correctness of $\boldsymbol{d}$ by using a product argument.

TWO-PROCESSOR SCHEDULING *(*2PS *for short) [GJ79, p. 65].* 2PS is the set of all vectors $(\boldsymbol{S} = (S_1, \ldots, S_n), s)$, such that there exist $\boldsymbol{b} \in \{0,1\}^n$ with $\sum_{i=1}^n S_i b_i \leq s$ and $\sum_{i=1}^n S_i(1 - b_i) \leq s$. To construct a short NIZK argument for 2PS, we show that the scan of $\boldsymbol{S} \circ \boldsymbol{b}$ is in the range $[0 \mathinner{..} s]$. Here, we use the range argument of App. I twice. One can similarly construct an NIZK argument for multiprocessor scheduling [GJ79, p. 65] for m processors, but the complexity of the argument will depend linearly on m.

KNAPSACK *[GJ79, p. 65].* KNAPSACK is the set of all vectors $(\boldsymbol{S} = (S_1, \ldots, S_n), \boldsymbol{V} = (V_1, \ldots, V_n), s, v)$, such that there exist $\boldsymbol{b} \in \{0,1\}^n$ with $\sum_{i=1}^n S_i b_i \leq s$ and $\sum_{i=1}^n V_i b_i \geq v$. To construct a short NIZK argument for KNAPSACK, we show that the scan of $\boldsymbol{S} \circ \boldsymbol{b}$ is in the range $[0 \mathinner{..} s]$ and the scan of $\boldsymbol{V} \circ \boldsymbol{b}$ is in the range of (say) $[v \mathinner{..} p - 1]$. Here, we use the range argument of App. I twice.

None of the mentioned languages is strongly NP-complete [GJ78], since they can be decided by polynomial-time algorithms given that all integers are small. Since the new arguments allow the integers to be exponential in κ, they work in the range where the underlying languages are infeasible. The adaptive CIRCUIT-SAT argument of [Lip14], when combined with the basic arguments of the current paper, is $\Theta(\log n)$ times less efficient. Whether one can construct adaptive arguments for strongly NP-complete languages without the factor $\Theta(\log n)$ slow-down is an interesting open question.