Almost Optimal Short Adaptive Non-Interactive Zero Knowledge

First eprint version, May 30, 2014

Helger Lipmaa

University of Tartu, Estonia

Abstract. Several recent short NIZK arguments are constructed in a modular way from a small number of basic arguments like the product argument or the shift argument. The main technical novelty of the current work is a significantly more efficient version of the product argument. Based on this, we propose an adaptive NIZK range argument with almost optimal complexity: constant communication (in group elements), constant verifier's computational complexity (in cryptographic operations), and $\Theta(n \log n)$ [resp., $\Theta(n)$] prover's computational complexity (in non-cryptographic [resp., cryptographic] operations). The latter can be compared to $n \log^{\omega(1)} n$ in the most efficient *published* short adaptive non-interactive range argument, or $\Theta(n \log^2 n)$ [resp., $\Theta(n \log n)$] that is achievable when following QAP-based framework from Eurocrypt 2013. Here, n is the logarithm of the range length. The new product argument can be used to construct efficient adaptive NIZK arguments for many other languages, including several that are NP-complete like Subset-Sum. Importantly, for all such languages, new adaptive arguments achieve better prover's computation than the QAP-based framework.

Keywords: CRS, NIZK, range proof, interpolation, product argument, quadratic arithmetic program

1 Introduction

A common approach to design a cryptographic protocol, secure in the malicious model, is to (i) design a protocol, secure in the semihonest model under the assumption that the committed or encrypted inputs are valid, and (ii) accompany it with a zero-knowledge proof [GMR85] that the inputs are indeed valid. Such an approach is common say in homomorphic e-voting [CGS97,DJ01], where the vote tally is only correct when every voter encrypted the number of a valid candidate. In particular, in many such applications it often suffices to prove — possibly for many inputs — that an input belongs to some publicly fixed range $[L\,..\,H] = \{L, L+1, \ldots, H-1, H\}$. That is, to use a *range proof*.

To understand important considerations that have to be taken into account when constructing range proofs, recall that by using a non-interactive zero knowledge (NIZK) proof [BFM88], the prover can create a short proof π that some claim is true, without revealing any side information. Since the proof is non-interactive, the same proof π can be forwarded to many different verifiers who can independently verify the truth of the claim. The latter property is important in many applications like e-voting or e-auctions, where one cannot trust the voter (resp., 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, that is, arguments. Ideally, an argument should consist only of a few (say) group elements. By the same reason, an argument should also be efficient to verify. On the other hand, the construction of the argument can be somewhat less efficient, since it is only done once. Still, prover-efficiency is important, for example in a situation where a single server has to create many arguments to different clients. All the mentioned considerations play an important role in the case of range proofs.

Related Work. Due to the importance of the problem, there has been a large quantity of work on range proofs, some of which use quite unexpected ideas from other branches of mathematics. In particular, there has been a large number of previous work on constructing interactive range 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]. Moreover, known range proofs in the random oracle model have either suboptimal communication (e.g., linear in the bitlength n of the range, with $n := \lfloor \log_2(H-L) \rfloor + 1$) or rather high computational complexity (e.g., in [Lip03], the communication is $\Theta(1)$ but the prover has to execute the randomized Rabin-Shallit algorithm [RS86] that takes quadratic time assuming the Extended Riemann Hypothesis). On the other hand, [Gro11] achieved $\Theta(n \log^2 n)$ prover's computation but with $\Theta(n^{1/3})$ communication. See [Bou00,LAN02,DGS02,Gro04,CCs08,CLs10] for just some more related work.

There are only a few short NIZK range arguments in the standard model, that is, in the common reference string (CRS, [BFM88]) model. First, [RKP09] proposed a range argument with communication of $\Theta(n/\log n)$ group elements. However since a group element is at least $\Theta(\log n)$ bits, the communication is not a sublinear number of bits. The first range argument in the standard model with constant communication was proposed in [CLZ12] and then made more efficient in [FLZ13]. More precisely, by following the pioneering work of Groth [Gro10], the range arguments from [CLZ12,FLZ13] are built up in a modular way from a small number of basic arguments. E.g., the range argument of [FLZ13] is 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 shift argument (given commitments to $\boldsymbol{a}$, $\boldsymbol{b}$, it holds that $\boldsymbol{a}$ is a coordinate-shift of $\boldsymbol{b}$; first proposed in [FLZ13]), and a small number of other arguments. As shown in [FLZ13], the same basic arguments can be used to construct NIZK arguments for other languages, including several NP-complete languages.

Interestingly, existing short range arguments in the standard model are quite efficient. For example, the product argument of [Lip12] and thus also the range argument of [CLZ12] has constant communication, constant verifier's computation, and quadratic prover's computation. In what follows, we count computation often implicitly in group elements, and verifier's computation in the number of cryptographic operations. However, in the case case of prover's computation (which is the least efficient part of the argument), usually the number of non-cryptographic operations and cryptographic operations differs.

Therefore, 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-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 by Straus [Str64] and Pippenger [Pip80]. The number of non-cryptographic operations in the prover's computation in the product argument of [Lip12] (and thus also in the range argument of [CLZ12]) can be decreased to $\Theta(r_3^{-1}(n) \log r_3^{-1}(n))$ by using the Fast Fourier Transform, see [FLZ13].

Finding explicit progression-free sets with the large r_3 (and thus the small r_3^{-1}) function is probably one of the best known classical hard problems in additive combinatorics (it is listed as the first classical open problem in [CL07]). By a recent breakthrough result 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 in practice a better choice is to choose the progression-free set of Erdős and Turán [ET36] with $r_3^{-1}(n) = n^{\log_3 2}$. In either case, the range argument of [CLZ12,FLZ13] has still better prover's computational complexity than the (random-oracle model) argument of [Lip03].

The prover's computational complexity of the range argument of [FLZ13] is strongly dominated by that of the product argument; in fact, the prover's computation in the rest of the range argument is linear in n. Hence, an important open question is to optimize the product argument of [Gro10,Lip12,FLZ13]. A more efficient product argument would result in a more efficient range argument, but also to more efficient arguments for many other languages. Such languages include NP-complete languages Set-Partition, Decision-Knapsack and Subset-Sum [FLZ13] and (although we leave it as an open question) possibly also Circuit-SAT or even verifiable computation [GGP10,GGPR13]. 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.

Our Contribution. One way to improve on the product argument of [FLZ13] is to construct a progression-free set that improves upon that of Elkin [Elk11]. However, since this is a well-known hard problem in additive combinatorics, we will instead avoid the use of progression-free sets, and use a different methodology due to [GGPR13]. The resulting new product argument requires prover's computation of $\Theta(n \log n)$ (versus

$\Theta(r_3^{-1}(n)\log r_3^{-1}(n))$ in [FLZ13]) non-cryptographic operations and $\Theta(n)$ (versus $\Theta(r_3^{-1}(n))$ in [FLZ13]) cryptographic operations. This also results in faster arguments for other languages that use the shift-and-product framework of [FLZ13], like Subset-Sum. Such efficiency would be impossible by using the progression-free set based approach due to the known upper bounds on $r_3(N)$ [San11,Blo14]. Moreover, the new adaptive arguments are asymptotically faster than the quadratic arithmetic program (QAP) based solutions of [GGPR13] (even when taking into accounts later improvements, proposed in [PGHR13,BSCG+13,Lip13]) for the same languages. A detailed comparison to direct QAP-based solutions is given in Sect. 7.

Nevertheless, the new product argument follows the QAP-based blueprint of [GGPR13]. The product argument is essentially a polynomial quadratic arithmetic program (QAP) for the circuit that computes n multiplications in parallel, with additional elements to guarantee security in our setting. The prover's computation in the new product argument is dominated by three polynomial interpolations, one polynomial multiplication and one polynomial division, all over $\mathbb{Z}_p$, and two n-wide multi-exponentiations. This takes total time $\Theta(n \log n)$, assuming that p satisfies a mild criterion.

The derivation of the new product argument automatically results in a new (homomorphic) trapdoor commitment scheme[1] — the *interpolating commitment scheme* — that is a variant of known commitment schemes known since at least [GJM02] (see [Gro10,KZG10,Lip12,FLZ13] for various generalizations). Its security proof is a modification of the security proofs of the latter. In fact, the interpolating commitment scheme is a very natural commitment scheme: commitment to a vector $\boldsymbol{a}$ is basically a short garbled version of the Lagrange interpolating polynomial $L_{\boldsymbol{a}}(X)$ of this vector, where $L_{\boldsymbol{a}}(X)$ can be computed by using inverse Fast Fourier Transform [CT65]. Thus, there are certain parallels between the new product argument and the well-known FFT-based multiplication algorithm.

We then construct a variant of the shift argument of [FLZ13] that is secure when combined with the interpolating commitment scheme; it has only one non-trivial change compared to [FLZ13]. We also show that the restriction argument of [Gro10] can be modified to work with the interpolating commitment scheme.

Based on the new trapdoor commitment scheme, the new product argument, the new shift argument, and the restriction argument, we describe a version of the range argument from [CLZ12,FLZ13]. The new range argument can be seen as a short program in the scan vector parallel computation model [Ble90] that operates on committed vectors of length n. The steps of this short program consist of one application of the restriction argument and of the shift argument, and two applications of the new product argument.

To emphasize that the new range argument is conceptually simple, we will next give its full description; it can be compared with the much more complicated argument of [CLZ12]. In particular, due to the efficient prover's computation, we do not have to consider various trade-offs, presented in [CLZ12] (though they are still available), but can propose one instantiation of the range argument that fares well in all parameters.

As usually, we assume that $L = 0$ and thus we need to construct a range argument that $a \in [0\,..\,H]$. (This is possible, since from $a \in [0\,..\,H]$ it follows that $a + L \in [L\,..\,H + L]$.) Let $n = \lfloor \log_2 H \rfloor + 1$ be the bit-length of H. We first reinterpret the commitment to $a \in [0\,..\,H]$ as a commitment to the n-dimensional vector $(0, \ldots, 0, a)$. We rely on a result from [LAN02] (formally proven in [CLs10]) that $a \in [0\,..\,H]$ iff $a = \sum_{i=1}^{n} H_i b_i$, where H_i are publicly known constants and $b_i \in \{0, 1\}$. We commit to $\boldsymbol{b}$, and use the new product argument to show that $b_i \in \{0, 1\}$. We commit to $\boldsymbol{c}$, where $c_i = H_i b_i$, and use the product argument to prove that $\boldsymbol{c}$ was computed correctly. We compute a commitment to $\boldsymbol{d}$, where $d_i = \sum_{j \geq i} c_j$, and use the shift argument to show that this was done correctly. We finally use a restriction argument from [Gro10] to show that $a = d_1$, that is, $a = \sum_{j=1}^{n} c_j = \sum_{j=1}^{n} H_j b_j \in [0\,..\,H]$. The security of the range argument follows from the security of the basic arguments and related knowledge assumptions.

In the new range argument, the prover's computation is $\Theta(n \log n)$ non-cryptographic operations (dominated by six polynomial interpolations, two polynomial multiplications and two polynomial divisions) and $\Theta(n)$ cryptographic operations (dominated by a small number of n-wide multi-exponentiations). Both parameters are significantly improved when compared to [FLZ13], and arguably (almost) optimal. In fact,

[1] It is not uncommon to propose range proofs, or NIZK arguments in general, based on non-standard commitment schemes. One example is the range proof of [Lip03] that is based on the integer commitment scheme of Damgård and Fujisaki [DF02]. Similarly, the arguments of [Gro10,Lip12,FLZ13] used tailored commitment schemes.

the prover's computation is strongly dominated by $\Theta(n)$ cryptographic operations.[2] The argument size is 18 group elements, and the verifier's computation is dominated by 32 bilinear pairings. The CRS consists of $\Theta(n)$ group elements. In addition, [CLZ12] proposed several variants of their range argument that offer various trade-offs between the computational and communication complexity. The same trade-offs can be used here, but the resulting range arguments are obviously more efficient. See [CLZ12] for more information.

Other Applications. A major benefit of the modular approach of [Gro10] is its generality: one can use the new product argument to speed up the prover's computation in NIZK arguments for other languages, including several NP-complete languages [FLZ13]. All such applications can be easily modified to use the interpolating commitment scheme, after that they require prover's computation of $\Theta(n \log n)$ non-cryptographic operations. This can be compared to $\Theta(n \log^2 n)$ when using the QAP-based approach directly. Similar speed-up is achieved in the case of cryptographic operations.

More generally, Fauzi et al [FLZ13] constructed an efficient vector scan argument. 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 flexible and powerful, and can be used to implement many problems efficiently. Constructing efficient arguments for other cryptographically relevant languages in this model is an interesting direction of future work. In particular, we leave it as an open problem to construct a similarly efficient argument for Circuit-SAT (and thus also for verifiable computation). For this it suffices to design an efficient permutation argument; see App. F for discussion.

2 New Trapdoor Commitment Scheme

In this section, we construct the 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. The precise reasoning behind the parameters will become clear in Sect. 4.

Trapdoor commitment scheme is a randomized cryptographic primitive in the CRS model [BFM88] that takes a message and outputs its commitment. It consists of two efficient algorithms gencom (that outputs a CRS and a trapdoor) and com (that, given the 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 the same commitment to two different messages. **Perfect hiding:** commitments of any two messages have the same distribution. **Trapdoor:** given an access to the original message, the randomizer and the trapdoor, one can open a commitment to (say) 0 to an arbitrary message. See, e.g., [Gro10] for formal definitions.

We define the following pairing-based *polynomial (trapdoor) commitment scheme*. Recall that on input 1^κ, where κ is the security parameter, the *bilinear map generator* [SOK00,Jou00,BF01] returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, g_1, g_2)$, where $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are three multiplicative cyclic groups of prime order p, g_z is a generator of $\mathbb{G}_z$ for $z \in \{1, 2\}$, and $\hat{e}$ is an efficient bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that satisfies in particular the following two properties: (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 $ab = cd \mod p$.

Definition 1 ($\{P_i\}$-Commitment Scheme). *Let $n = \mathrm{poly}(\kappa)$, $n > 0$, be an integer, and let $z \in \{1, 2\}$. Let $P_i(X) \in \mathbb{Z}_p[X]$, for $i \in \{0, 1, \ldots, n\}$, be distinct linearly independent low-degree polynomials. The $\{P_i\}$-commitment scheme is parametrized by $(z, n, \{P_i\}_{i=0}^n)$. First, $\mathsf{gencom}(1^\kappa, n)$ invokes the bilinear group generator to generate $\mathsf{gk} \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, g_1, g_2)$, and outputs the CRS*

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

[2] Some speed-ups might still be possible, since cryptographic operations are executed with $\Omega(\log n)$-bit operands. E.g., in [CCs08,RKP09,CLs10], the prover's computation is $\Theta(n/\log n)$ cryptographic operations, but the argument size is also $\Theta(n/\log n)$ group elements and thus the argument is not short. We leave combining techniques from their argument with the ones from the current paper an interesting open question.

for uniform and random $\alpha \leftarrow_r \mathbb{Z}_p$ *and* $\sigma \leftarrow_r \mathbb{Z}_p \setminus \{x : P_0(x) = 0\}$. *The trapdoor is equal to* (σ, α).

Second, the commitment $\mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r)$ *of* $\boldsymbol{a} \in \mathbb{Z}_p^n$, *given a fresh randomizer* $r \leftarrow_r \mathbb{Z}_p$, *is equal to* $(g_z^{P_0(\sigma)}, g_z^{\alpha P_0(\sigma)})^r \cdot \prod_{i=1}^n (g_z^{P_i(\sigma)}, g_z^{\alpha P_i(\sigma)})^{a_i} \in \mathbb{G}_z^2$. *The validity of a commitment* (A, A^α) *can be checked by verifying that* $\hat{e}(A, g_2^{\alpha P_0(\sigma)}) = \hat{e}(A^\alpha, g_2^{P_0(\sigma)})$ *(if* $z = 1$*) or* $\hat{e}(g_1^{\alpha P_0(\sigma)}, A) = \hat{e}(g_1^{P_0(\sigma)}, A^\alpha)$ *(if* $z = 2$*). To open a commitment, the committer sends* $(\boldsymbol{a}, r)$ *to the verifier.*

Since a commitment consists of two group elements, for a commitment (A, A^α), let us denote $\mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r) = (\mathsf{com}_1(\mathsf{ck}; \boldsymbol{a}; r), \mathsf{com}_2(\mathsf{ck}; \boldsymbol{a}; r))$. Clearly,

$$\log_{g_z} A = rP_0(\sigma) + \sum_{i=1}^n a_i P_i(\sigma) \ . \tag{2}$$

The second element, A^α, of the commitment is known as the knowledge component [Dam91].

The security of the $\{P_i\}$-commitment scheme (and thus also of the arguments of the current paper) depends on the following q-type assumptions, variants of which have been studied and used in say [GJM02,DL08,Gro10,CLZ12,Lip12,BCI+13,FLZ13]. All known (to us) adaptive short NIZK arguments are based on q-type assumptions about genbp. Informally (see App. A for formal definitions that in particular give syntactic restrictions to gen), let gen be an efficient algorithm that on input $(1^\kappa, n)$ generates a CRS crs and a trapdoor $(\sigma, \ldots, \alpha, \ldots)$. Let gen-DL be the computational assumption that given crs output by $\mathsf{gen}(1^\kappa, n)$, it is difficult to compute σ. The gen-DL assumption is a variant of the uber-assumption from [BBG05]. It is possible that it could be simplified, but we leave it as an open problem. See [CM14]. Let $(\mathsf{gen}, \{P_i\}, \alpha)$-KE be the knowledge assumption [Dam91] that a committer who has created a valid commitment (A, A^α) in $\mathbb{G}_z$, given $\mathsf{ck} \leftarrow \mathsf{gen}(1^\kappa, n)$ (but not the trapdoor) as input, must know coefficients a_i such that $\log_{g_z} A = \sum a_i P_i(\sigma)$. The following theorem together with its security proof is standard.

Theorem 1. *Let* $z \in \{1, 2\}$. *Let* $\Phi_{\mathsf{com}} = \{P_i\}_{i=0}^n$, *where* P_i *are as in Def. 1. The* $\{P_i\}$*-commitment scheme is perfectly hiding. It is computationally binding when* genbp *is* gencom*-DL secure. If* genbp *is* $(\mathsf{gencom}, \{P_i\}, \alpha)$*-KE secure, then there exists an extractor that extracts the message* $\boldsymbol{a}$ *and the randomizer* r, *given* ck, *the commitment* $(A, A^\alpha) = \mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r)$ *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 $\log_{g_z} A$) is uniformly random in $\mathbb{Z}_p$. Therefore, (A, A^α) is a uniformly random element of the multiplicative subgroup of $\mathbb{G}_z^2$ generated by (g_z, g_z^α), 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 x_i of $d(X)$. One of these roots has to be equal to σ (one can establish which one by comparing each (say) $g_z^{P_0(x_i)}$ to the element $g_z^{P_0(\sigma)}$ given in the CRS).[3] □

See App. B 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 the* $\{P_i\}$*-commitment scheme, instantiated with polynomials* $P_0(X) = Z(X)$ *and* $P_i(X) = \ell_i(X)$ *for* $i \in \{1, \ldots, n\}$ *over* $\mathbb{Z}_p[X]$, *that are 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,*

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

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

Thus also $\Phi_{\mathsf{com}} = \{Z(X)\} \cup \{\ell_i(X)\}_{i=1}^{n}$.

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

In the case when ω_i are arbitrary, polynomial interpolation can be done in time $\Theta(n\log^2 n)$ [GG03]. This slows down the prover's computation to $\Theta(n\log^2 n)$ non-cryptographic operations in the new range argument, and to $\Theta(n\log^3 n)$ in a QAP-based argument. Note that for the existence of the n-th primitive root of unity modulo p it suffices that $(n+1) \mid (p-1)$. One can then use the Cocks-Pinch method 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.

This precise choice of the polynomials $Z(X)$ and $\ell_i(X)$ will be motivated later, in Sect. 4. It is easy to see that 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), 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 arguments.

3 Preliminaries: Zero Knowledge

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

An NIZK argument for a language $\mathcal{L}$ consists of three algorithms, gencrs, prove and ver. The CRS generation algorithm gencrs takes as input 1^κ (and possibly some other, public, language-dependent information) and outputs the prover's CRS crs_p, the verifier's CRS crs_v, and the 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 prove takes as an input crs_p together with a statement x and a witness w, and outputs an argument π. The verifier ver takes as an input crs_v together with a statement x and an argument π, and either accepts or rejects.

Some of the expected properties of an argument are: (i) perfect completeness (the honest verifier always accepts the honest prover), (ii) perfect witness-indistinguishability (the distributions of the arguments corresponding to any two allowable witnesses are the same), (iii) perfect zero knowledge (there exists an efficient simulator that can, given x, crs_p and td, output an argument that comes from the same distribution as the argument produced by the prover), and (iv) computational soundness (if $x \notin \mathcal{L}$, then an arbitrary nonuniform probabilistic polynomial time prover has only a negligible success in creating a satisfying argument). 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. D.

4 Hadamard Product Argument

Here and in what follows, $\boldsymbol{a} \circ \boldsymbol{b}$ denotes the Hadamard (element-wise) multiplication of two vectors, with $(\boldsymbol{a} \circ \boldsymbol{b})_i = a_i b_i$. In an *product argument* [Gro10], the prover aims to convince the verifier that she knows how to open three commitments A, B, and C to vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ correspondingly, such that $\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{c}$. We first follow the line of thought of [GGPR13] (but using the matrix notation of [Lip13]) to derive the new product argument together with the supporting trapdoor commitment scheme. After that, we give a full description of the argument together with a discussion of its security and efficiency.

The underlying commitment scheme will be the interpolating commitment scheme. Quick intuition behind this is that the commitment scheme stores a garbled version of the interpolating polynomial $L_{\boldsymbol{a}}$ of

the input vector $\boldsymbol{a}$. The polynomial $L_{\boldsymbol{a}}$ can be computed in $\Theta(n \log n)$ non-cryptographic operations, and garbling takes $\Theta(n)$ cryptographic operations. Since multiplication of two vectors, given their interpolating polynomials, can be done pointwise in time $\Theta(n)$, we can in $\Theta(n \log n)$ time compute the polynomial $Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X) = L_{\boldsymbol{a}}(X)L_{\boldsymbol{b}}(X) - L_{\boldsymbol{c}}(X)L_{\mathbf{1}_n}(X)$, and from $Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X)$ we can compute the actual argument in $\Theta(n)$ cryptographic operations. Given this interpretation, we could have omitted several steps in the next derivation, but we omitted not to do it for the sake of clarity.

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

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

which in turn holds if and only if

$$(\sum a_i \boldsymbol{e_i}) \circ (\sum b_i \boldsymbol{e_i}) = (\sum c_i \boldsymbol{e_i}) \circ (\sum \boldsymbol{e_i}) \ . \tag{4}$$

Following the terminology of [Lip13], Eq. (4) 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 $\boldsymbol{c}$ as the output. Importantly, in Eq. (3) we use the same matrix, I_n, in the matrix-vector products $\boldsymbol{a}^\top I_n$, $\boldsymbol{b}^\top I_n$, and $\boldsymbol{c}^\top I_n$. At the end, this will mean that we can commit to three vectors $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ by using the $\{P_i\}$-commitment scheme with the same parameters as in Def. 2.

Next, from Eq. (4) we obtain a *polynomial quadratic arithmetic program* for $\mathcal{C}$ as in [Lip13]. Fix n different values $(\omega_1, \ldots, \omega_n)$ as in Def. 2. Now, let $\ell_i(X)$ be again the ith Lagrange basis polynomial, as in Def. 2. Clearly, due to the definition of $\ell_i(X)$ (and recalling that $L_{\boldsymbol{x}}(X) = \sum x_i \ell_i(X)$), Eq. (4) is equivalent to the requirement that the degree $n-2$ polynomial

$$Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X) := L_{\boldsymbol{a}}(X)L_{\boldsymbol{b}}(X) - L_{\boldsymbol{c}}(X) \cdot L_{\mathbf{1}_n}(X)$$

evaluates to 0 at all n values ω_i. Thus, for $Z(X)$ defined as in Def. 2, Eq. (4) is equivalent to $Z(X) \mid Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X)$. That is, there exists a degree $(2n-2) - n = n - 2$ polynomial $\pi(X)$, such that

$$\pi(X) \cdot Z(X) = Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X) \ . \tag{5}$$

Since $\boldsymbol{c} = \boldsymbol{a} \circ \boldsymbol{b}$, $\pi(X)$ can be computed as $\pi(X) \leftarrow Q^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{a}\circ\boldsymbol{b}}(X)/Z(X)$.

Finally, 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

$$Q_{zk}^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X) := (L_{\boldsymbol{a}}(X) + r_a Z(X))\,(L_{\boldsymbol{b}}(X) + r_b Z(X)) - (L_{\boldsymbol{c}}(X) + r_c Z(X)) \cdot L_{\mathbf{1}_n}(X) \ .$$

Here, the added elements of type $r_a Z(X)$ guarantee hiding. On the other hand, due to the use of $Z(X)$ in addends, since $Z(\omega_i) = 0$, $Q_{zk}^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X)$ 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 there exists a polynomial $\pi_{zk}(X)$ such that

$$\pi_{zk}(X) \cdot Z(X) = Q_{zk}^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X) \ . \tag{6}$$

Here, the degree $n-2$ polynomial $\pi_{zk}(X)$ can be computed as

$$\pi_{zk}(X) := Q_{zk}^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{a}\circ\boldsymbol{b}}(X)/Z(X) \ . \tag{7}$$

Moreover, $\pi_{zk}(X)$ does not reveal any information about the witness.

However, $\pi_{zk}(X)$ is not of sublinear length in n. As common in situations like that, to minimize communication, we instead transfer the evaluation of $\pi_{zk}(X)$ at a random secret point σ. In particular, $g_1^{Q_{zk}^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(\sigma)} := (\mathsf{com}_1(\mathsf{ck}; \boldsymbol{a}; r_a) \cdot \mathsf{com}_1(\mathsf{ck}; \boldsymbol{b}; r_b)) \,/\, (\mathsf{com}_1(\mathsf{ck}; \boldsymbol{c}; r_c) \cdot \mathsf{com}_1(\mathsf{ck}; \mathbf{1}; 0))$, where we use the interpolating commitment scheme from Def. 2 with correctly defined ck.

However, since σ is an unknown secret element, the prover cannot compute $\pi_{zk}(\sigma)$. Instead, he computes $g_1^{\pi_{zk}(\sigma)}$, using the values $g_1^{\sigma^i}$ (given in the CRS) and the coefficients π_i of $\pi_{zk}(X) = \sum_{i=0}^{n-2} \pi_i X^i$ (computed according to Eq. (7)), together with its knowledge component (here, β is another secret key), as follows:

$$(\pi, \pi^\beta) := (g_1, g_1^\beta)^{\pi_{zk}(\sigma)} \leftarrow \prod_{i=0}^{n-2} (g_1^{\sigma^i}, g_1^{\beta\sigma^i})^{\pi_i} \ . \tag{8}$$

Hadamard Product Argument: Details. The now give a detailed description of the new product argument. Note that we partition the CRS into the prover's CRS crs_p and the verifier's CRS crs_v. This is not important for the security (both parties may get access to both CRS-s), but since crs_p is significantly shorter, it may result in some gain of efficiency in practice.

CRS generation $\mathsf{gencrs}_\times(1^\kappa, n)$**:** Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, g_1, g_2) \leftarrow \mathsf{genbp}(1^\kappa)$. Let com be the interpolating commitment scheme from Def. 2. Generate $(\sigma, \alpha, \beta) \leftarrow_r \mathbb{Z}_p^3$ with $Z(\sigma) \neq 0$. Set $\mathsf{ck}_z \leftarrow (\mathsf{gk}; g_z^{f(\sigma)}, g_z^{\alpha f(\sigma)})_{f \in \{Z, \ell_1, \ldots, \ell_n\}}$ for $z \in \{1, 2\}$. Set $E \leftarrow \mathsf{com}_1(\mathsf{ck}_2; \mathbf{1}_n; 0) = \prod_{i=1}^n (g_2^{\ell_i(\sigma)}, g_2^{\alpha \ell_i(\sigma)})$. Finally, set $\mathsf{crs}_p \leftarrow (\mathsf{ck}_1, \mathsf{ck}_2, (g_1^{f(\sigma)}, g_1^{\beta f(\sigma)})_{f \in \{X^0, \ldots, X^{n-2}\}})$, $\mathsf{crs}_v \leftarrow (\mathsf{gk}; g_2^{Z(\sigma)}, g_2^\beta, E)$, and $td \leftarrow (\sigma, \alpha, \beta)$. Output $(\mathsf{crs} = (\mathsf{crs}_p, \mathsf{crs}_v), td)$.

Proving $\mathsf{prove}_\times(\mathsf{crs}_p; (A, A^\alpha, B, B^\alpha, C, C^\alpha; w_\times = (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{c}, r_c))$**:** Compute $\pi_\times \leftarrow (\pi, \pi^\beta)$ as by Eq. (8). Output $\pi_\times$.

Verification $\mathsf{ver}_\times(\mathsf{crs}_v; A, A^\alpha, B, B^\alpha, C, C^\alpha; \pi_\times)$**:** Verify that (i) $\hat{e}(\pi, g_2^\beta) = \hat{e}(\pi^\beta, g_2)$, and (ii) $\hat{e}(A, B) = \hat{e}(C, E) \cdot \hat{e}(\pi, g_2^{Z(\sigma)})$.

Security. Like any of the other basic arguments, the product argument cannot be sound by the standard definition of soundness, see [Gro10,Lip12] for a detailed explanation. It can only satisfy a weaker notion of soundness, which basically states that if there exists an adversary that creates all the elements that include the knowledge component (that is, all such inputs and the argument itself), then one can construct an extractor that extracts the witness (in this case, $w_\times$). This weaker version of soundness is however sufficient for (say) the range 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 (say) the range argument to be zero-knowledge.

Theorem 2. *Let $n = \mathrm{poly}(\kappa)$. Let com be the interpolating commitment scheme from Def. 2. Assume that the inputs are valid commitments.*

1. *The new product argument is perfectly complete and perfectly witness-indistinguishable.*
2. *(*Weak soundness:*) If genbp is $\mathsf{gencrs}_\times$-DL secure, then a non-uniform probabilistic polynomial-time adversary against the new product argument has negligible chance, given $\mathsf{crs} \leftarrow \mathsf{gencrs}(1^\kappa, n)$ as an input, of outputting $inp_\times = (A, A^\alpha, B, B^\alpha, C, C^\alpha)$ and an accepting argument $\pi_\times = (\pi, \pi^\beta)$ together with a witness $w_\times = (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{c}, r_c, \boldsymbol{\pi'} = (\pi'_i)_{i=0}^{n-2})$, such that*
   (i) $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{Z}_p^n$, $r_a, r_b, r_c \in \mathbb{Z}_p$, and $\boldsymbol{\pi'} \in \mathbb{Z}_p^{n-1}$,
   (ii) $(A, A^\alpha) = \mathsf{com}(\mathsf{ck}_1; \boldsymbol{a}; r_a)$, $(B, B^\alpha) = \mathsf{com}(\mathsf{ck}_2; \boldsymbol{b}; r_b)$, and $(C, C^\alpha) = \mathsf{com}(\mathsf{ck}_1; \boldsymbol{c}; r_c)$,
   (iii) $\log_{g_1} \pi = \log_{g_1^\beta} \pi^\beta = \sum_{i=0}^{n-2} \pi'_i \sigma^i$, and
   (iv) for some $i \in [1 \,..\, n]$, $a_i b_i \neq c_i$.

Proof. Completeness: follows from the derivation of the argument 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 therefore the product argument is witness-indistinguishable.

Weaker version of soundness. Assume that $\mathcal{A}$ is an adversary that can break the last statement of the theorem. We construct an adversary $\mathcal{A}_{dl}$ against the $\mathsf{gencrs}_\times$-DL assumption. Let $(\mathsf{crs}, td) \leftarrow \mathsf{gencrs}(1^\kappa, n)$. The adversary $\mathcal{A}_{dl}$ receives crs as her input, and her task is to output σ. She sends crs to $\mathcal{A}$.

Assume that $\mathcal{A}$ returns $(inp_\times, \pi_\times, w_\times)$ such that the conditions in the theorem statement hold, and $\mathsf{ver}_\times(\mathsf{crs}; inp_\times; \pi_\times)$ accepts. Here, $inp_\times = (A, A^\alpha, B, B^\alpha, C, C^\alpha)$ and $w_\times = (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{c}, r_c, \boldsymbol{\pi'})$.

If $\mathcal{A}$ is successful, then $\mathcal{A}_{dl}$ has recovered all the coefficients of the polynomial $Q_{zk}^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X)$ and also all the coefficients of the polynomial $\pi_{zk}(X)$. Moreover, due to the discussion in the beginning of this section, $Q_{zk}^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X)$ evaluates to 0 at all n values ω_i (that is, $Z(X) \mid Q_{zk}^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X)$) if and only if $\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{c}$. Since $\mathcal{A}$ was successful, it must be the case that $d(X) := \pi(X) \cdot Z(X) - Q_{zk}^{\boldsymbol{a},\boldsymbol{b},\boldsymbol{c}}(X)$ is a non-trivial polynomial, for which $\mathcal{A}_{dl}$ knows all the coefficients.

Since the last verification equation holds, it means that $d(\sigma) = 0$. The adversary can now apply an efficient polynomial factorization algorithm in $\mathbb{Z}_p[X]$ [LLL82,vHN10] to find all roots x_i of $d(X)$. One of the

roots has to be equal to σ; $\mathcal{A}_{dl}$ can find the correct root by comparing $g_2^{\ell_j(\sigma)}$ for some j, given in the CRS, with all values $g_2^{\ell_j(x_i)}$. The adversary has thus violated the $\mathsf{gencrs}_\times$-DL assumption. □

Efficiency. The prover's computation is dominated by the computation of (i) two multi-exponentiations of width $n-1$. By using the Pippenger's multi-exponentiation algorithm [Pip80], this means asymptotically approximately $2(n-1)$ multiplications in a bilinear group. For small values of n, one can use the algorithm by Straus [Str64]. (ii) three polynomial interpolations, one polynomial multiplication (since $L_{\mathbf{1}}(X)=1$, one multiplication can be omitted) and one polynomial division to compute the coefficients of the polynomial $\pi_{zk}(X)$. Since polynomial division can be implemented as two 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. C.

The verifier's computation is dominated by 5 pairings. Excluding gk, the prover's CRS consists of $4n$ group elements, while the verifier's CRS consists of only 3 group elements. The CRS can be computed in linear time, by using the algorithm proposed in [BSCG+13].

5 On Other Basic Arguments

In the new range argument we will be relying on the product argument of Sect. 4 but also on two other arguments, the shift argument from [FLZ13] and the restriction argument from [Gro10]. We will rederive the shift argument (for the interpolating commitment scheme) and briefly describe the restriction argument. See the original papers for more details.

5.1 Right Shift-by-ξ Argument

In a *right shift-by-ξ* argument [FLZ13], the prover aims to convince the verifier that for two commitments A and B, he knows how to open them as $A=\mathsf{com}(\mathsf{ck};\boldsymbol{a};r_a)$ and $B=\mathsf{com}(\mathsf{ck};\boldsymbol{b};r_b)$, such that $a_i=b_{i+\xi}$ for $i\in[1\,..\,n-\xi]$ and $a_i=0$ for $i\in[n-\xi+1\,..\,n]$. That is, $(a_n,\ldots,a_1)=(0,\ldots,0,b_n,\ldots,b_{1+\xi})$.

An efficient right shift-by-ξ argument was described in [FLZ13]. We now reconstruct the right shift argument of [FLZ13] so that it can be used together with the interpolating commitment scheme of Def. 2. While we only need the case $\xi=1$, the general case is as easy to handle as the special case.

There are several reasons why one can design an efficient right shift argument that works together with the interpolating commitment scheme. Most importantly, the right shift argument of [FLZ13] is very efficient to start with, and the construction of the argument of [FLZ13] almost does not depend on the concrete commitment scheme. We mention that the argument needs one non-trivial modification compared to [FLZ13]: as we see in what follows, for the security reasons we set a certain polynomial $\zeta(X)$ to be equal to $Z(X)$, while in [FLZ13], $\zeta(X)$ had a different definition $\zeta(X)=X^\xi$.

Our strategy of constructing the shift argument follows the strategies 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 $F(X)$ and $\pi_{honest}(X)$, each of which belongs to the span of a set of polynomials. The second polynomial, $\pi_{honest}(X)$, is identically zero if and only if the prover is honest. Under the assumption that the spans of two polynomial sets do not intersect, this results in the right shift argument.

Generalizing [FLZ13], assume that the verification equation is $\hat{e}(B\cdot g_1^{\pi(\sigma)},g_2)=^?\hat{e}(A,g_2^{\zeta(\sigma)})$, for A and B being commitments to $\boldsymbol{a}$ and $\boldsymbol{b}$ (by using the $\{P_i\}$-commitment scheme, without fixing the polynomials yet), and $\zeta(X)$ being a polynomial that we will fix later. The value $g_1^{\pi(\sigma)}$ corresponds to the argument. Denote $R(X):=r_aP_0(X)\zeta(X)-r_bP_0(X)$. Replacing σ with a formal variable X and taking a discrete logarithm of

the verification equation,

$$\begin{aligned}\pi(X) &= \left(r_a P_0(X) + \sum a_i P_i(X)\right)\zeta(X) - \left(r_b P_0(X) + \sum b_i P_i(X)\right) \\ &= \sum a_i P_i(X)\zeta(X) - \sum b_i P_i(X) + R(X) \\ &= \left(\sum_{i=1}^{n-\xi} a_i P_i(X)\zeta(X) + \sum_{i=n-\xi+1}^{n} a_i P_i(X)\zeta(X)\right) - \left(\sum_{i=1}^{n-\xi} b_{i+\xi} P_{i+\xi}(X) + \sum_{i=1}^{\xi} b_i P_i(X)\right) + R(X) \\ &= F(X) + \pi_{honest}(X) \ ,\end{aligned}$$

where

$$\begin{aligned}F(X) &= \left(\sum_{i=1}^{n-\xi}(a_i - b_{i+\xi})P_i(X)\zeta(X) + \sum_{i=n-\xi+1}^{n} a_i P_i(X)\zeta(X)\right) \ , \\ \pi_{honest}(X) &= \left(\sum_{i=1}^{n-\xi} b_{i+\xi}(P_i(X)\zeta(X) - P_{i+\xi}(X)) - \sum_{i=1}^{\xi} b_i P_i(X)\right) + R(X) \ .\end{aligned}$$

Now, if the prover is honest, then $F(X) = 0$, and thus $\pi(X) = \pi_{honest}(X)$ belongs to the span of $\Phi^{\xi}_{\mathsf{rsft}} := \{P_i(X)\zeta(X) - P_{i+\xi}(X)\}_{i=1}^{n-\xi} \cup \{P_i\}_{i=1}^{\xi} \cup \{P_0(X)\zeta(X)\} \cup \{P_0(X)\}$. For the argument to be sound, we need that $P_i(X)$, $i \geq 1$, are all linearly independent, and that $F(X) \notin span(\Phi^{\xi}_{\mathsf{rsft}})$, that is, $P_k(X)\zeta(X) \notin span(\Phi^{\xi}_{\mathsf{rsft}})$, for $k \in [1\,..\,n]$. (This guarantees that from a representation of $\pi(X)$ as an element of $span(\Phi^{\xi}_{\mathsf{rsft}})$ 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 $\zeta(X)$.

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

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

It holds that $\ell_k(X)Z(X) \notin span(\Phi^{\xi}_{\mathsf{rsft}})$ for any $k \in [1\,..\,n]$.

Proof. Assume that for some $k \in [1\,..\,n]$, $\ell_k(X)Z(X) \in span(\Phi^{\xi}_{\mathsf{rsft}})$. First, $Z(X)^2$ is the only polynomial of degree $\geq 2(n+1)$ and therefore can be "removed" from the span. Thus, there exit integers a_i, b_i and c_i, such that $\ell_k(X)Z(X) = aZ(X) + \sum_{i=1}^{n-\xi} b_i\,(\ell_i(X)Z(X) - \ell_{i+\xi}(X)) + \sum_{i=1}^{\xi} c_i\ell_i(X)$. But then the left hand side and the right hand side polynomials must also agree on points ω_i, for $i \in [1\,..\,n]$. Therefore, due to the definition of the olynomials ℓ_i and Z, $\mathbf{0} = -\sum_{i=1}^{n-\xi} b_i \boldsymbol{e_{i+\xi}} + \sum_{i=1}^{\xi} c_i \boldsymbol{e_i} = \sum_{i=1}^{\xi} c_i \boldsymbol{e_i} + \sum_{i=\xi+1}^{n} b_{i-\xi}\boldsymbol{e_i}$. The latter is only possible if $b_i = c_i = 0$. Since $\ell_k(X)Z(X) \neq aZ(X)$ for constant a, this finishes the proof. □

As in the case of the product argument, the argument will not contain the polynomial $\pi(X)$ itself, but the value $(g_1^{\pi(\sigma)}, g_1^{\gamma\pi(\sigma)})$ for random σ and a knowledge secret γ, computed as

$$\begin{aligned}(g_1^{\pi(\sigma)}, g_1^{\gamma\pi(\sigma)}) = &\prod_{i=1}^{n-\xi}\left(g_1^{\ell_i(\sigma)Z(\sigma)-\ell_{i+\xi}(\sigma)}, g_1^{\gamma(\ell_i(\sigma)Z(\sigma)-\ell_{i+\xi}(\sigma))}\right)^{b_{i+\xi}} \cdot \prod_{i=1}^{\xi}\left(g_1^{\ell_i(\sigma)}, g_1^{\gamma\ell_i(\sigma)}\right)^{-b_i} \cdot \\ &\left(g_1^{Z(\sigma)^2}, g_1^{\gamma Z(\sigma)^2}\right)^{r_a}\left(g_1^{Z(\sigma)}, g_1^{\gamma Z(\sigma)}\right)^{-r_b} \ .\end{aligned} \tag{9}$$

We are now ready to state the full right-shift-by-ξ argument:

CRS generation $\mathsf{gencrs}_{\mathsf{rsft}}(1^\kappa, n)$**:** Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, g_1, g_2) \leftarrow \mathsf{genbp}(1^\kappa)$. Generate $(\sigma, \alpha, \gamma) \leftarrow \mathbb{Z}_p^3$ with $Z(\sigma) \neq 0$. Set $\mathsf{ck}_1 \leftarrow (g_1^{f(\sigma)}, g_1^{\alpha f(\sigma)})_{f \in \{Z, \ell_1, \ldots, \ell_n\}}$. Let $\mathsf{crs}_p \leftarrow (\mathsf{gk}; g_1^{f(\sigma)}, g_1^{\gamma f(\sigma)})_{f \in \Phi^{\xi}_{\mathsf{rsft}}})$ (this contains ck_1), $\mathsf{crs}_v \leftarrow (\mathsf{gk}; g_2^{Z(\sigma)})$. Set $td \leftarrow (\sigma, \alpha, \gamma)$. Return $((\mathsf{crs}_p, \mathsf{crs}_v), td)$.

Proving $\mathsf{prove}_{\mathsf{rsft}}(\mathsf{crs}_p; A, A^\alpha, B, B^\alpha; \boldsymbol{a}, r_a, \boldsymbol{b}, r_b)$**:** compute $\pi_{\mathsf{rsft}} \leftarrow (\pi, \pi^\gamma)$ as in Eq. (9). Return π_{rsft}.

Verification $\mathsf{ver}_{\mathsf{rsft}}(\mathsf{crs}_v; A, A^\alpha, B, B^\alpha; \pi_{\mathsf{rsft}} = (\pi, \pi^\gamma))$**:** check that (i) $\hat{e}(g_1^{Z(\sigma)}, \pi^\gamma) = \hat{e}(g_1^{\gamma Z(\sigma)}, \pi)$ ((π, π^γ) is valid), and (ii) $\hat{e}(B \cdot \pi, g_2) = \hat{e}(A, g_2^{Z(\sigma)})$.

Theorem 3. *Let $n = \mathrm{poly}(\kappa)$. Let* com *be the commitment scheme from Def. 2. Assume that the inputs are valid commitments.*

1. *The shift argument of [FLZ13] is perfectly complete and perfectly witness-indistinguishable.*
2. *(*Weak soundness:*) Let $\Phi^{\xi}_{\mathsf{rsft}}$ be as in Lem. 1. If* genbp *is* $\mathsf{gencrs}_{\mathsf{rsft}}$*-DL secure, then a non-uniform probabilistic polynomial time adversary against the shift argument of the current section has negligible chance, given $\mathsf{crs} \leftarrow \mathsf{gencrs}(1^\kappa, n)$ as an input, of outputting $inp_{\mathsf{rsft}} \leftarrow (A, A^\alpha, B, B^\alpha)$ and an accepting argument (π, π^γ) together with a witness $w_{\mathsf{rsft}} \leftarrow (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, (f^*_\varphi)_{\varphi \in \Phi^{\xi}_{\mathsf{rsft}}})$, such that*
   - *(i) $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{Z}_p^n$, $r_a, r_b \in \mathbb{Z}_p$, $f^*_\varphi \in \mathbb{Z}_p$ for $\varphi \in \Phi^{\xi}_{\mathsf{rsft}}$,*
   - *(ii) $(A, A^\alpha) = \mathsf{com}(\mathsf{ck}; \boldsymbol{a}; r_a)$, $(B, B^\alpha) = \mathsf{com}(\mathsf{ck}; \boldsymbol{b}; r_b)$,*
   - *(iii) $\log_{g_2} \pi = \log_{g_2^\beta} \pi^\gamma = \sum_{\varphi \in \Phi^{\xi}_{\mathsf{rsft}}} f^*_\varphi \cdot \varphi(\sigma)$, and*
   - *(iv) $(a_n, \ldots, a_1) \neq (0, \ldots, 0, b_n, \ldots, b_{\xi+1})$.*

(See App. E for a proof sketch.)

Efficiency. The prover's computation is dominated by two $(n+1)$-wide multi-exponentiations. Note that this time there is no need for polynomial interpolation, multiplication or division. The communication is 2 group elements. The verifier's computation is dominated by 4 pairings.

5.2 Restriction Argument

In a *restriction argument* [Gro10], the prover aims to convince the verifier that some entries of the committed vector $\boldsymbol{a}$ are equal to $\mathbf{0}$. That is, for a publicly known index set $I \subseteq [1\,..\,n]$, it follows from $i \in I$ that $a_i = 0$. Groth proposed a simple restriction argument in [Gro10]. Assume one uses the $\{P_i\}$-commitment scheme, where $\{P_i\}$ satisfy the requirements of Thm. 1 (e.g., they come from Def. 2), and that $z \in \{1, 2\}$. The basic idea behind this argument is that for a secret knowledge element $\delta \leftarrow_r \mathbb{Z}_p$ (that is specific to the restriction algorithm), only the values $(g_z, g_z^\delta)^{P_i(\sigma)}$, where $i \in \{0\} \cup I$, are available as a part of the CRS. The knowledge component A^δ of the commitment to $\boldsymbol{a}$, $(A, A^\delta) \leftarrow (g_z^{P_0(\sigma)}, g_z^{\delta P_0(\sigma)})^r \cdot \prod_{i=1}^n (g_z^{P_i(\sigma)}, g_z^{\delta P_i(\sigma)})^{a_i}$ is then essentially equal to its own restriction argument (a single $(n+1)$-wide multi-exponentiation, and 1 group element). The verification consists of checking that (A, A^δ) is a valid commitment (2 pairings).

We note that the restriction argument can be implemented using a product argument, but the described direct implementation is more efficient. As a drawback, inclusion of a direct restriction argument means putting more elements to the CRS of the (say) range argument.

6 Range Argument

In a range argument, given public range $[L\,..\,H]$, the prover aims to convince the verifier that he knows how to open the commitment A to a value $a \in [L\,..\,H]$. Next, we show that by using the new product argument, one can design a range argument with a significantly better prover's computation than it was known before. The new range argument is similar to one version of the range argument of [CLZ12,FLZ13], but it is simpler due to the use of the new product argument. 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 $H \geq 1$.

Construction. Assume that the common input (A, A^α) is a commitment to the vector $\boldsymbol{a}$ with $a_0 = a$ and $a_i = 0$ for $i > 0$. To prove that $a \in [0\,..\,H]$ for some H and $n = \lfloor \log_2 H \rfloor + 1$, we do the following.

The CRS generation $\mathsf{gencrs}_{\mathsf{range}}$ invokes the CRS generations of the commitment scheme, the two restriction arguments, the product argument and the shift argument, sharing the same gk and trapdoor $td = (\sigma, \alpha, \beta, \gamma, \delta_1, \delta_2)$ between the different invocations.

The prover does the following (further explanations are given after the argument itself):

Construct another commitment (A_1, A_1^α) of $\boldsymbol{a}$ in group $\mathbb{G}_1$.
Construct a restriction argument $\pi_1^{\delta_1}$ (knowledge component of A/A_1) that A/A_1 commits to $\mathbf{0}$.
Let $a = \sum_{i=1}^n H_i b_i$ for $H_i = \left\lfloor (H + 2^{i-1})/(2^i) \right\rfloor$ and $b_i \in \{0, 1\}$.
For $z \in \{1, 2\}$, let (B_z, B_z^α) be a commitment to $\boldsymbol{b}$ in group $\mathbb{G}_z$.
Construct a product argument (π_2, π_2^β) to show that $\boldsymbol{b} = \boldsymbol{b} \circ \boldsymbol{b}$.
Let (C, C^α) be a commitment to $\boldsymbol{c}$ in group $\mathbb{G}_1$, where $c_i = H_i b_i$.
Construct a product argument (π_3, π_3^β) to show that $\boldsymbol{c} = \boldsymbol{H} \circ \boldsymbol{b}$.
Let (D, D^α) be a commitment to $\boldsymbol{d}$ in group $\mathbb{G}_1$, where $d_i = \sum_{j \geq i} c_i$.
Construct a shift argument (π_4, π_4^γ) to show that $\boldsymbol{d} - \boldsymbol{c}$ is a right shift of $\boldsymbol{d}$
Construct a restriction argument $\pi_5^{\delta_2}$ to show that $(A_1/D, A_1^\alpha/D^\alpha)$ commits to $\boldsymbol{f}$ with $f_1 = 0$.
Output $\pi_{\mathsf{range}} = (A_1, A_1^\alpha, B_1, B_1^\alpha, B_2, B_2^\alpha, C, C^\alpha, D, D^\alpha, \pi_1^{\delta_1}, \pi_2, \pi_2^\beta, \pi_3, \pi_3^\beta, \pi_4, \pi_4^\gamma, \pi_5^{\delta_2})$.

After receiving π_{range}, the verifier checks the validity of six commitments (A, A^α), (A_1, A_1^α), (B_1, B_1^α), (B_2, B_2^α), (C, C^α), and (D, D^α), verifies that $\hat{e}(B_1, g_2) = \hat{e}(g_1, B_2)$, and then verifies the five arguments.

The vector $\boldsymbol{d}$ is called either a *vector scan*, an *all-prefix-sums* or a *prefix-sum* of $\boldsymbol{c}$ [Ble90], and π_4 can be thought of a *scan argument* [FLZ13] that $\boldsymbol{d}$ is a correct scan of $\boldsymbol{c}$. Moreover, [CLZ12] also considered the case where a was encrypted by the BBS cryptosystem [BBS04], and gave an NIZK argument that the same value of a has been encrypted and been committed to. If needed, we can use the same additional argument to establish that an *encrypted* value belongs to a certain range.

We will now give the security claim, listing precisely all used KE assumptions. It is easy to see that all used assumptions satisfy the syntactic requirements given in App. A.

Theorem 4. *Let $n = \mathrm{poly}(\kappa)$, and let com be the interpolating commitment scheme from Def. 2. Let $\Phi_{rest_1} = \{Z(X)\}$ and $\Phi_{rest_2} = (\{Z(X)\} \cup \{\ell_i(X)\}_{i=2}^n)$. The new range argument is perfectly complete, computationally sound, and perfectly zero-knowledge. It is computationally sound and an argument of knowledge if genbp satisfies the following assumptions: the $\mathsf{gencrs}_{\mathsf{range}}$-DL assumption, the $(\mathsf{gencrs}_{\mathsf{range}}, \Phi_{rest_1}, \delta_1)$, $(\mathsf{gencrs}_{\mathsf{range}}, \Phi_{\mathsf{com}}, \alpha)$, $(\mathsf{gencrs}_{\mathsf{range}}, \{X^i\}_{i=1}^n, \beta)$, $(\mathsf{gencrs}_{\mathsf{range}}, \Phi_{\mathsf{rsft}}^1, \gamma)$, and $(\mathsf{gencrs}_{\mathsf{range}}, \Phi_{rest_2}, \delta_2)$ KE assumptions in $\mathbb{G}_1$ and $(\mathsf{gencrs}_{\mathsf{range}}, \Phi_{\mathsf{com}}, \alpha)$-KE assumption in $\mathbb{G}_2$.*

Proof (Sketch). Completeness: note that $a \in [0\,..\,H]$ 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_2 proves that b_i are Boolean, π_3 proves that $c_i = H_i b_i$, π_4 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_5 proves that $a - d_1 = a - \sum_{i=1}^n H_i b_i = 0$. Thus, $a = \sum_{i=1}^n H_i b_i$ and therefore, $a \in [0\,..\,H]$.

Computational soundness follows, under the corresponding DL and KE assumptions on genbp, from the weak soundness of every basic argument. First, note that if genbp is $\mathsf{gencrs}_{\mathsf{range}}$-DL secure, then also the gen^*-DL assumption holds for gen^* being any of the constituent CRS generations (e.g., for $\mathsf{gen}^* = \mathsf{gencrs}_\times$). Therefore, from the $\mathsf{gencrs}_{\mathsf{range}}$-DL assumption it follows that every subargument is weakly sound.

In the proof of soundness (that follows the idea of the proof from [CLZ12]), assume that the corresponding KE assumptions hold and that there exists an adversary $\mathcal{A}$ that breaks the soundness of the range argument. One then construct an adversary $\mathcal{A}_{dl}$ that breaks the corresponding gen^*-DL assumption on genbp as follows. First of all, by the completeness of the range argument, the adversary $\mathcal{A}$ must have broken one of the basic arguments. Say this argument is π_3 (the other cases are analogous). By the knowledge assumptions on genbp (in this case, $(\mathsf{gencrs}_{\mathsf{range}}, \Phi_{\mathsf{com}}, \alpha)$-KE and $(\mathsf{gencrs}_{\mathsf{range}}, \{X^i\}_{i=0}^n, \beta)$-KE in $\mathbb{G}_1$ and $(\mathsf{gencrs}_{\mathsf{range}}, \Phi_{\mathsf{com}}, \alpha)$-KE in $\mathbb{G}_2$), $\mathcal{A}_{dl}$ can obtain all committed values touched in this argument (in this case, $\boldsymbol{c}$ and $\boldsymbol{b}$) together with the representation of the discrete logarithm of the basic argument as a linear function of type $\sum_{i=0}^n \pi_i X^i$. But

then it follows from the weak soundness of the product argument that $\mathcal{A}_{dl}$ has broken $\mathsf{gencrs}_{\mathsf{range}}$-DL security of genbp.

Let gencrs_{rest_i} be the CRS generation function that corresponds to the ith restriction argument (i.e., to either π_1 when $i = 1$, or to π_5 when $i = 2$). Then,

(i) since genbp is $(\mathsf{gencrs}_{\mathsf{range}}, \Phi_{\mathsf{com}}, \alpha)$-KE secure in $\mathbb{G}_1$ and $\mathbb{G}_2$, there exists an extractor that obtains $\boldsymbol{a_1}$, $\boldsymbol{b}$, $\boldsymbol{c}$ and $\boldsymbol{d}$ (and the used randomizers r_{a_1}, r_b, r_c and r_d) from the commitments (A_1, A_1^α), (B, B^α), (C, C^α), and (D, D^α).

(ii) since genbp is $(\mathsf{gencrs}_{\mathsf{range}}, \Phi_{rest_1}, \delta_1)$-KE secure in $\mathbb{G}_1$, there exists an extractor that obtains $\boldsymbol{a'} = \boldsymbol{a} - \boldsymbol{a_1}$ (and the used randomizers) from the argument π_1. Thus, $\mathcal{A}_{dl}$ has access to all values required in (a slight modification of) Thm. 1, and hence by the gencrs_{rest_1}-DL assumption, $\boldsymbol{a} = \boldsymbol{a_1}$.

(iii) since genbp is $(\mathsf{gencrs}_{\mathsf{range}}, \{X^i\}_{i=0}^n, \beta)$-KE secure in $\mathbb{G}_1$, there exists an extractor that obtains a linear representation $\pi_2 = \sum_{i=0}^n \pi_i X^i$ from the argument (π_2, π_2^β). Thus, $\mathcal{A}_{dl}$ has access to all values required in Thm. 2, and hence by the $\mathsf{gencrs}_\times$-DL assumption, $b_i \in \{0, 1\}$.

(iv) since genbp is $(\mathsf{gencrs}_{\mathsf{range}}, \{X^i\}_{i=0}^n, \beta)$-KE secure in $\mathbb{G}_1$, there exists an extractor that obtains a linear representation $\pi_3 = \sum_{i=0}^n \pi_i X^i$ from the argument (π_3, π_3^β). Thus, $\mathcal{A}_{dl}$ has access to all values required in Thm. 2, and hence by the $\mathsf{gencrs}_\times$-DL assumption, $\boldsymbol{c} = \boldsymbol{H} \circ \boldsymbol{b}$.

(v) since genbp is $(\mathsf{gencrs}_{\mathsf{range}}, \Phi^1_{\mathsf{rsft}}, \gamma)$-KE secure in $\mathbb{G}_1$, there exists an extractor that obtains both $\boldsymbol{c}$ and $\boldsymbol{d}$ (and the used randomizers) from the argument (π_4, π_4^γ). Thus, $\mathcal{A}_{dl}$ has access to all values required in Thm. 3, and hence by the $\mathsf{gencrs}_{\mathsf{rsft}}$-DL assumption, $d_i = \sum_{j \geq i} c_j = \sum_{j \geq i} H_j b_j$.

(vi) since genbp is $(\mathsf{gencrs}_{\mathsf{range}}, \Phi_{rest_2}, \delta_2)$-KE secure in $\mathbb{G}_1$, there exists an extractor that obtains $\boldsymbol{f} = \boldsymbol{a} - \boldsymbol{d}$ (and the used randomizers) from the argument π_5. Thus, $\mathcal{A}_{dl}$ has access to all values required in (a slight modification of) Thm. 1, and hence by the gencrs_{rest_2}-DL assumption, $f_1 = 0$ and thus $a_1 = d_1$.

Argument of knowledge: follows the above proof of soundness.

Perfect zero-knowledge: follows from the presence of the trapdoor. The simulator basically creates (A_1, A_1^α), (B, B^α), (C, C^α), (D, D^α) as commitments to $\mathbf{0}$, except that in the first argument she computes $(A_1, A_1^\alpha) \leftarrow (A, A^\alpha) \cdot \mathsf{com}(\mathsf{ck}_1; \mathbf{0}; r)$ for a random r. She then simulates the basic arguments, based on her knowledge of the trapdoor. All five arguments are obviously correct when the committed values are equal to $\mathbf{0}$: $\boldsymbol{a} - \boldsymbol{a} = \mathbf{0}$ (this takes care of π_1), $\mathbf{0} = \mathbf{0} \circ \mathbf{0}$ (this takes care of π_2), $\mathbf{0} = \boldsymbol{H} \circ \mathbf{0}$ (this takes care of π_3), $\mathbf{0}$ is a right shift of $\mathbf{0}$ (this takes care of π_4), and $\mathbf{0} = \mathbf{0}$ (this takes care of π_5). □

The commitment (A_1, A_1^α) and the restriction argument $\pi_1^{\delta_1}$ are only necessary to achieve simulatability in the case A is a part of the common input (by following an idea from [FLZ14]). It can ommitted when the prover actually creates (A, A^α). However, this step adds some additional flexibility to the proof: for example, one can instead of (A, A^α) using an encryption of $\boldsymbol{a}$ and then use a somewhat less efficient argument that A encrypts to the same values as A_1 decrypts to, see [CLZ12].

Efficiency. The prover's computation is dominated by the application of two product arguments. Thus, the prover's computational complexity is dominated by $\Theta(n \log n)$ non-cryptographic operations and $\Theta(n)$ cryptographic operations. The argument size is constant (18 group elements), and the verifier's computational complexity is dominated by 32 pairings (5 pairings in either product argument, 4 pairings in the shift argument, 2 pairing in either restriction argument, 2 pairings to verify that B_1 and B_2 commit to the same element, and 12 pairings to verify the validity of 6 commitments).

The resulting range argument is hence significantly more computationally efficient than the previous arguments [CLZ12,FLZ13]. In fact, it has also better communication (18 versus 31 group elements in [FLZ13]), and verification complexity (32 versus 65 pairings in [FLZ13]). Moreover, it is also simpler: since the prover's computation is quasilinear, we do not have to consider various trade-offs between computation and communication as in [CLZ12,FLZ13]. (These trade-offs are still available, if needed.)

7 Comparison to Direct QAP-Based Argument

We will now give a brief comparison with a possible direct QAP-based range argument. The QAP-based range argument that we outline next will be more efficient than the previous *published* range arguments but not as efficient than the range argument of the current paper.

Recall that one can use the techniques of [GGPR13] to construct an adaptive NIZK argument to show that a circuit $\mathcal{C}$ that, given (secret) input $\boldsymbol{a}$, outputs publicly known value (say 1). To use these techniques, we can first design an arithmetic circuit that outputs 1 iff the input is in the range (since this circuit needs to execute an integer comparison, it will have size $\Theta(n)$), and then use the conversions of [GGPR13,BCI$^+$13] to design an NIZK range argument with very short argument (7 group elements). However, the direct construct results in an non-adaptive argument. This means in particular, that — by using the terminology of the current paper — the inputs will be committed by the $\{P_i\}$-commitment scheme, where the choice of the polynomials P_i depends on the concrete circuit (in the case of range argument, on H). Such a dependence is undesirable in many applications, since it means that a separate CRS has to be regenerated by a trusted party for every particular application . In the case of range arguments, this means that a separate CRS has to be created for every particular H; e.g., in e-voting CRS can only generated after the number of candidates is known.

Thus, in many applications it is highly desirable to have an a priori fixed commitment scheme that is known to all participants and does not depend on the concrete application. As mentioned in [GGPR13], the dependence on the concrete circuit can be avoided by using universal circuits [Val76] that take the original circuit as one of its inputs; this results in an adaptive NIZK argument. Valiant's universal circuit has size $\Theta(n \log n)$, where n is the original circuit size. However, this means that the polynomials P_i depend on the construction of the universal circuit, which is not desirable for efficiency and compatibility reasons. (Constructions based on universal circuits rarely get implemented; this is partially due to the huge constants involved in the $O(n \log n)$ expression. E.g. [KS08,SS08] have implemented asymptotically less efficient but in-practice better universal circuits.) Moreover, composing the range argument with some other arguments can become problematic due to possible incompatibilities between the underlying commitment schemes.

The new interpolating commitment scheme of the current paper depends on the arithmetic circuit for the Hadamard product. This makes it easier to compose it with other NIZK arguments that are constructed by using the vector scan model. In particular, the CRS can be generated once, and then used in many different NIZK arguments. Finally, the new range argument does not depend on universal circuits, and is thus more efficient — by a factor of $\Theta(\log n)$ — than a direct application of the QAP-based techniques of [GGPR13]. The latter would namely require prover's computation of $\Theta(n \log^2 n)$ non-cryptographic and $\Theta(n \log n)$ cryptographic operations.

Importantly, the current work seems be the first application where the QAP-based techniques of [GGPR13] are combined with unrelated techniques to create a more efficient NIZK argument.

8 Application: Other Arguments

One can use the new basic arguments to also design a number of other arguments by using different techniques. We discuss Circuit-SAT in App. F.

Product and Shift Framework. As shown in [FLZ13], arguments for other interesting languages can be constructed, given efficient arguments for Hadamard product and shift. This includes NP-complete languages like Subset-Sum, Decision-Knapsack and Set-Partition. One can plug in the interpolating commitment scheme and the new product argument to speed up corresponding arguments. For example, also the Subset-Sum argument in [FLZ13] consists of product, shift, and restriction argument. Interestingly, when instantiated with the interpolating commitment scheme, all such arguments will have prover's computation dominated by $\Theta(n \log n)$ non-cryptographic operations and $\Theta(n)$ cryptographic operations, where n is an argument-dependent parameter. Here, n is some language-dependent parameter (e.g., the size of the big set in Subset-Sum). It is unknown how to achieve similar efficiency by using any other techniques.

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

References

Ano14. Anonymized. Efficient NIZK Arguments via Parallel Verification of Benes Networks. Under submission, 2014.

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.

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short Group Signatures. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55, Santa Barbara, USA, August 15–19, 2004. Springer, Heidelberg.

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.

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.

BG92. Mihir Bellare and Oded Goldreich. On Defining Proofs of Knowledge. In Ernest F. Brickell, editor, *CRYPTO 1992*, volume 740 of *LNCS*, pages 390–420, Santa Barbara, California, USA, August 16–20, 1992. Springer, Heidelberg, 1993.

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`.

Bou00. Fabrice Boudot. Efficient Proofs That a Committed Number Lies in an Interval. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 431–444, Bruges, Belgium, May 14–18, 2000. Springer, Heidelberg.

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.

CCs08. Jan Camenisch, Rafik Chaabouni, and abhi shelat. Efficient Protocols for Set Membership and Range Proofs. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 234–252, Melbourne, Australia, December 7–11, 2008. Springer, Heidelberg.

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.

CGS97. Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In Walter Fumy, editor, *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 103–118, Konstanz, Germany, 11–15 May 1997. Springer, Heidelberg.

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.

CM14. Melissa Chase and Sarah Meiklejohn. Déjà Q: Using Dual Systems to Revisit q-Type Assumptions. In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 622–639, Copenhagen, Denmark, May 11–15, 2014. Springer, Heidelberg.

CT65. James W. Cooley and John W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19:297–301, 1965.

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.

DF02. Ivan Damgård and Eiichiro Fujisaki. An Integer Commitment Scheme Based on Groups with Hidden Order. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 125–142, Queenstown, New Zealand, December 1–5, 2002. Springer, Heidelberg.

DGS02. Ivan Damgård, Jens Groth, and Gorm Salomonsen. *The Theory and Implementation of an Electronic Voting System*, pages 77–99. Kluwer Academic Publishers, 2002.

DJ01. Ivan Damgård and Mads Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 119–136, Cheju Island, Korea, February 13–15, 2001. Springer, Heidelberg.

DL08. Giovanni Di Crescenzo and Helger Lipmaa. Succinct NP Proofs from an Extractability Assumption. In Arnold Beckmann, Costas Dimitracopoulos, and Benedikt Löwe, editors, *Computability in Europe, CIE 2008*, volume 5028 of *LNCS*, pages 175–185, Athens, Greece, June 15–20, 2008. 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.

FLZ14. Prastudy Fauzi, Helger Lipmaa, and Bingsheng Zhang. Efficient Non-Interactive Zero Knowledge Arguments for Set Operations. In Nicolas Christin and Rei Safavi-Naini, editors, *FC 2014*, volume ? of *LNCS*, pages ?–?, Bridgetown, Barbados, March 3–7, 2014. 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.

GGP10. Rosario Gennaro, Craig Gentry, and Bryan Parno. Non-Interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 465–482, Santa Barbara, California, USA, August 15–19, 2010. Springer, Heidelberg.

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.

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.

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.

Gro04. Jens Groth. *Honest Verifier Zero-Knowledge Arguments Applied*. PhD thesis, University of Århus, Denmark, October 2004.

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.

Gro11. Jens Groth. Efficient Zero-Knowledge Arguments from Two-Tiered Homomorphic Commitments. In Dong Hoon Lee and Xiaoyun Wang, editors, *ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 431–448, Seoul, South Korea, December 4–8, 2011. 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.

JR13a. Charanjit Jutla and Arnab Roy. Switching Lemma for Bilinear Tests and Constant-size NIZK Proofs for Linear Subspaces. Technical Report 2013/670, International Association for Cryptologic Research, 2013. Available at `http://eprint.iacr.org/2013/670`, last accessed version from 17 Feb 2014.

JR13b. 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.

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.

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

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.

RKP09. Alfredo Rial, Markulf Kohlweiss, and Bart Preneel. Universally Composable Adaptive Priced Oblivious Transfer. In Hovav Shacham and Brent Waters, editors, *Pairing 2009*, volume 5671 of *LNCS*, pages 231–247, Palo Alto, CA, USA, August 12–14, 2009. Springer, Heidelberg.

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 Security Assumptions

Some of the predecessor papers have gone into great details in writing down the precise security assumptions, see for example [CLZ12,FLZ13]. We feel that this would sidetrack the reader from understanding the con-

structions, and have therefore opted to use the following simpler (more general) versions of the assumptions. From those, one can easily recover more precise versions.

Let gen be an algorithm that on inputs $(1^\kappa, n)$ returns (crs, td). In particular, crs includes an output of a bilinear group generator genbp. This bilinear group generator genbp is gen-*DL (discrete logarithm) secure* if for any non-interactive probabilistic polynomial-time adversary $\mathcal{A}$, the following probability is negligible in κ:

$$\Pr\left[(\mathsf{crs}, td = (\sigma, \ldots)) \leftarrow \mathsf{gen}(1^\kappa, n) : \mathcal{A}(\mathsf{crs}) = \sigma \right] .$$

In our applications, gen has several restrictions. First of all, $td = (\sigma, \alpha_1, \ldots, \alpha_m)$. Second, the only elements of crs that depend on σ have form $g_z^{f_j(\sigma)}$ or $g_z^{\alpha_i f_j(\sigma)}$ for publicly known argument-dependent polynomials f_j. In such cases, gen-DL is a variant of the power symmetric discrete logarithm assumptions, studied in say [GJM02,Gro10,Lip12,CLZ12]. All such assumptions are a variation 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 $\Phi = \{P_i\}_{i=0}^n$ be a tuple of polynomials. Similarly, genbp is $(\mathsf{gen}, \Phi, \alpha)$-*KE (knowledge of exponent) secure in group* $\mathbb{G}_z$ if for any non-interactive probabilistic polynomial-time $\mathcal{A}$ there exists an non-interactive probabilistic polynomial-time extractor $X_\mathcal{A}$, such that the following probability is negligible in κ:

$$\Pr\left[(\mathsf{crs}, td = (\sigma, \ldots, \alpha, \ldots)) \leftarrow \mathsf{gen}(1^\kappa, n), \left(c, \hat{c}; (a_i)_{i\in[0,n]}\right) \leftarrow (\mathcal{A}||X_\mathcal{A})(\mathsf{crs}) : \hat{c} = c^\alpha \wedge c \neq \prod_{i=0}^n g_z^{a_i P_i(\sigma)} \right] .$$

Again, to relate this assumption to earlier assumptions like the PKE assumptions in [Gro10,Lip12,CLZ12], we have to introduce several syntactic requirements on gen. More precisely, let $(\mathsf{crs}, td = (\sigma, \ldots, \alpha, \ldots) \leftarrow \mathsf{gen}(1^\kappa, n)$. We need that crs contains $(g_z^x, g_z^{\alpha x})$ only when $x = f(\sigma)$ for $f \in \Phi$. Moreover, there is no other element of crs that depends on α.

When the required syntactic requirements are fulfilled (in the applications we have this is clearly the case), then by generalizing [Gro10,Lip12], one can show that both the gen-DL and $(\mathsf{gen}, \{P_i\}, \alpha)$-KE assumptions hold in the generic group model.

B 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]. In [Lip12], the authors considered the case of $P_0(X) = 1$ and $P_i(X) = X^{\lambda_i}$, $i \in [1\,..\,n]$, for general λ_i 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 operations (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.

Gentry 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 $\mathcal{C}(\boldsymbol{x}) = y$ for public y, where $\boldsymbol{x}$ 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 construction of the universal circuit.

C Product Argument: Computation of $\boldsymbol{\pi}$

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 $\pi(X) = L_{\boldsymbol{a}}(X) \cdot L_{\boldsymbol{b}}(X) - L_{\boldsymbol{a}\circ\boldsymbol{b}}(X)L_{\boldsymbol{1}}(X)$. Recall that $\omega_j = \omega^{j-1}$. We 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 $j \in \{0,\ldots,2n-1\}$. Since $L_{\boldsymbol{a}}(\omega^j) = a_j$ for $j \in \{0,\ldots,n-1\}$ 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 $j \in \{n,\ldots,2n-1\}$, 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 $L_{\boldsymbol{a}\circ\boldsymbol{b}}(X)L_{\boldsymbol{1}}(X)$ 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.
(b) $L_{\boldsymbol{1}}(X) = 1$, so one polynomial multiplication can be omitted, and we are done.
3. Compute $\pi(X)$ by coordinate-wise subtraction.

D More on Zero Knowledge

History/Motivations. NIZK proofs [BFM88] allow the prover to convince the verifier that an input x 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 trusted setups 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. 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]. Many verifiers can then later independently verify the proof, by having access to the same CRS. The proof has to be complete, sound and satisfy the zero-knowledge property. 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 [JR13b] for 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 $\mathcal{R} = \{(C,w)\}$ be an efficiently computable binary relation with $|w| = \mathrm{poly}(|C|)$. Here, C is a statement, and w is a witness. Let $\mathcal{L} = \{C : \exists w, (C,w) \in \mathcal{R}\}$ be an **NP**-language. Let $n = |C|$ 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 probablistic polynomial-time algorithms: a common reference string (CRS) generator gencrs, a prover prove, and a verifier ver. For $(\mathsf{crs}_p, \mathsf{crs}_v) \leftarrow \mathsf{gencrs}(1^\kappa, n)$, $\mathsf{prove}(\mathsf{crs}_p; C, w)$ produces an argument π, and $\mathsf{ver}(\mathsf{crs}_v; C, \pi)$ outputs either 1 (accept) or 0 (reject).

Π is *perfectly complete*, if for all $n = \mathrm{poly}(\kappa)$,

$$\Pr[(\mathsf{crs}_p, \mathsf{crs}_v) \leftarrow \mathsf{gencrs}(1^\kappa, n), (C,w) \leftarrow \mathcal{R}_n : \mathsf{ver}(\mathsf{crs}_v; C, \mathsf{prove}(\mathsf{crs}_p; C, w)) = 1] = 1 \ .$$

Π is *computationally sound*, if for all $n = \mathrm{poly}(\kappa)$ and non-uniform probablistic polynomial-time $\mathcal{A}$,

$$\Pr[(\mathsf{crs}_p, \mathsf{crs}_v) \leftarrow \mathsf{gencrs}(1^\kappa, n), (C,\pi) \leftarrow \mathcal{A}(\mathsf{crs}_p, \mathsf{crs}_v) : C \notin \mathcal{L} \wedge \mathsf{ver}(\mathsf{crs}_v; C, \pi) = 1] = \mathrm{negl}(\kappa) \ .$$

Π is *perfectly witness-indistinguishable*, if for all $n = \mathrm{poly}(\kappa)$, if $(\mathsf{crs}_p, \mathsf{crs}_v) \in \mathsf{gencrs}(1^\kappa, n)$ and $((C,w_0),(C,w_1)) \in \mathcal{R}_n^2$, then the distributions $\mathsf{prove}(\mathsf{crs}_p; C, w_0)$ and $\mathsf{prove}(\mathsf{crs}_p; C, w_1)$ are equal. Π is *perfectly zero-knowledge*, if there exists a probablistic polynomial-time simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$, such that for all stateful non-uniform probablistic polynomial-time adversaries $\mathcal{A}$ and $n = \mathrm{poly}(\kappa)$ (with td_π being the *simulation trapdoor*),

$$\Pr\begin{bmatrix}(\mathsf{crs}_p, \mathsf{crs}_v) \leftarrow \mathsf{gencrs}(1^\kappa, n), \\ (C,w) \leftarrow \mathcal{A}(\mathsf{crs}_p, \mathsf{crs}_v), \\ \pi \leftarrow \mathsf{prove}(\mathsf{crs}_p; C, w) : \\ (C,w) \in \mathcal{R}_n \wedge \mathcal{A}(\pi) = 1\end{bmatrix} = \Pr\begin{bmatrix}(\mathsf{crs}_p, \mathsf{crs}_v; td_\pi) \leftarrow \mathcal{S}_1(1^\kappa, n), \\ (C,w) \leftarrow \mathcal{A}(\mathsf{crs}_p, \mathsf{crs}_v), \\ \pi \leftarrow \mathcal{S}_2(\mathsf{crs}_p, \mathsf{crs}_v; C, td_\pi) : \\ (C,w) \in \mathcal{R}_n \wedge \mathcal{A}(\pi) = 1\end{bmatrix} \ .$$

For a formal definition of an argument of knowledge, see [BG92].

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 C is often called *non-adaptive*. See [JR13b] for a formalization of *quasi-adaptive* arguments. It is not surprising that non-adaptive (or quasi-adaptive) arguments are often much more efficient than adaptive arguments, see [GGPR13,JR13b,JR13a].

E Proof of Thm. 3

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.

(Weak) soundness: since the proof is very similar to the one in [FLZ13], we only sketch it. Assume that the adversary outputs the inputs, the witness and the coefficients of $\pi(X)$. We now construct the following extractor. It knows all coefficients in the Eq. (9), and thus has obtained coefficients of a polynomial $d(X)$, such that $d(\sigma) = 0$. If the prover was dishonest, then $d(X)$ is a non-zero polynomial. In this case the extractor can use a polynomial factorization algorithm to find all roots x_i of $d(X)$. One of those roots has to be σ; this can be tested by comparing (say) the values $g_2^{Z(x_i)}$ with the value $g_2^{Z(\sigma)}$ given in the CRS. □

F Open Problem: Circuit-SAT And Verifiable Computation

The prover's computational complexity of the Circuit-SAT argument of [Gro10,Lip12] is dominated by the prover's computational complexity of the product argument (quadratic in [Gro10,Lip12], $\Theta(r_3^{-1}(n) \log r_3^{-1}(n))$ in [FLZ13]) and the permutation argument (quadratic in [Gro10]). We proposed a product argument with prover's computation $\Theta(n \log n)$. Can one similarly improve the permutation argument? If so, one would automatically have an adaptive NIZK argument for Circuit-SAT, but also adaptive verifiable computation [GGP10], both with prover's computation $\Theta(n \log n)$ instead of $\Theta(n \log^2 n)$ in [GGPR13].

We have been informed by a *partial* process in the later question [Ano14], where the authors construct a permutation argument by using $\Theta(\log n)$ invocations of the product argument and the shift argument by using techniques related to Beneš networks. Combining ideas from [Ano14] and the current paper results in an adaptive Circuit-SAT argument (and adaptive verifiable computation) with prover's computation $\Theta(n \log^2 n)$ but with somewhat suboptimal communication $\Theta(\log n)$.

We leave constructing a more efficient permutation argument as an interesting problem.