Shorter Pairing-based Arguments under Standard Assumptions

Alonso González[1]* and Carla Ràfols[2]**

[1] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France.
[2] Universitat Pompeu Fabra and Cybercat, Barcelona, Spain.

Abstract. This paper constructs efficient non-interactive arguments for correct evaluation of arithmetic and boolean circuits with proof size $O(d)$ group elements, where d is the multiplicative depth of the circuit, under falsifiable assumptions. This is achieved by combining techniques from SNARKs and QA-NIZK arguments of membership in linear spaces. The first construction is very efficient (the proof size is $\approx 4d$ group elements and the verification cost is $\approx 4d$ pairings and $O(n + n' + d)$ exponentiations, where n is the size of the input and n' of the output) but one type of attack can only be ruled out assuming the knowledge soundness of QA-NIZK arguments of membership in linear spaces. We give an alternative construction which replaces this assumption with a decisional assumption in bilinear groups at the cost of approximately doubling the proof size. The construction for boolean circuits can be made zero-knowledge with Groth-Sahai proofs, resulting in a NIZK argument for circuit satisfiability based on falsifiable assumptions in bilinear groups of proof size $O(n + d)$.

Our main technical tool is what we call an "argument of knowledge transfer". Given a commitment C_1 and an opening x, such an argument allows to prove that some other commitment C_2 opens to $f(x)$, for some function f, even if C_2 is not extractable. We construct very short, constant-size, pairing-based arguments of knowledge transfer with constant-time verification for any linear function and also for Hadamard products. These allow to transfer the knowledge of the input to lower levels of the circuit.

1 Introduction

This paper deals with the problem of constructing non-interactive publicly verifiable arguments of knowledge under falsifiable assumptions to prove that a circuit ϕ is correctly evaluated in two different settings.

In one such possible setting, all of the input of the circuit ϕ is known. In this case, the argument does not need to be zero-knowledge and can leak partial

* This author was supported in part by the French ANR ALAMBIC project (ANR-16-CE39-0006).

** The research leading to this article was supported by a Marie Curie "UPF Fellows" Postdoctoral Grant and by Project RTI2018-102112-B-I00 (AEI/FEDER,UE)

information. This is the typical situation in verifiable computation in which a resource-limited device delegates a costly computation to a more powerful machine.

Another important setting requires the input and output to be partially or totally hidden and the argument to be zero-knowledge. This is interesting from a theoretical perspective as CircuitSat is usually taken to be the standard NP complete problem. On the practical side, often the best way to prove a large, complicated statement in zero-knowledge is to encode it as a circuit and prove that it is satisfiable. Further, CircuitSat is considered a sort of benchmark to evaluate the efficiency of zero-knowledge proofs.

Succinct Non-Interactive Arguments of Knowledge or SNARKs in bilinear groups have been a phenomenal success in both of these scenarios [15,28,8,1,16]. These arguments are succinct, more specifically, they are constant size, that is, not dependent on the circuit size, and extremely efficient also concretely (3 group elements in the best constructions [16]). They are also very fast to verify, which is a very interesting feature in practice, as in many scenarios verification is performed many times. However, these constructions still suffer from some problems, like long trusted parameters, heavy computation for the prover and reliance on non-falsifiable computational assumptions. Further, it is a well-known fact that the latter is unavoidable for succinct arguments in the non-interactive setting [11].

Non-falsifiable assumptions offer great efficiency at the price of less understood security guarantees. The problem is that it is not possible to efficiently check if the adversary effectively breaks the assumption, which results in non-explicit security reductions [32] which inherently do not allow to choose concrete security parameters meaningfully. Therefore, it is interesting to construct arguments with properties similar to SNARKs (short proof size, fast verification) for correct circuit evaluation that avoid falsifiable assumptions.

When the input of the circuit is public, SNARKs can be used to prove that the circuit is correctly evaluated while avoiding falsifiable assumptions. Indeed, since it is possible to check if a prover breaks soundness (as the input is public), the tautological assumption “the scheme is sound” is already falsifiable. For the case where at least some part of the input is secret, the same trivial solution can be used if the prover additionally commits to the input with some commitment which is extractable under falsifiable assumptions.[3] However, these trivial solutions require circuit dependent assumptions.

The goal of this paper is to design efficient constructions both in terms of proof size and verification complexity from milder (falsifiable, circuit independent) assumptions.

[3] Essentially the only such commitment known is bit to bit encryption, e.g. Groth-Sahai commitments to bits.

1.1 Our Results

We construct an argument for proving that an arithmetic circuit $\phi : \mathbb{Z}_p^n \to \mathbb{Z}_p^{n'}$ is correctly evaluated. We give two instantiations, the first one with proof size $(3d+2)\mathbb{G}_1 + (d+2)\mathbb{G}_2$ and where verification requires $4d+6$ parings and $O(n+n'+d)$ exponentiations, for d the depth of the circuit. We give a less efficient scheme where both proof size and verification cost are approximately the double of the first construction, more concretely, the proof size is $(6d+3)\mathbb{G}_1+(2d+3)\mathbb{G}_2$ and the verification requires $8d+9$ pairings.

For the first construction, we need to rely on the knowledge soundness of QA-NIZK arguments of membership in linear spaces, which has only been proven in the generic group model [5]. The second argument is fully based on falsifiable assumptions. The first one is an assumption that falls into the Matrix Decisional Diffie-Hellman assumption framework of Escala et al. [4] extended in asymmetric groups, where the challenge matrix is given in both groups. The size of the matrix depends on q, for q being the maximum number of multiplicative gates with the same multiplicative depth in the circuit. The second assumption is also a q-type assumption and similar to the q-SFrac Assumption of [12].

For boolean circuits, the argument can be made zero-knowledge with $O((n - n_{pub}) + d)$ proof size, where n_{pub} is the public input size.

1.2 Our Techniques

Circuit Satisfiability can be represented as a set of quadratic and linear equations. It would seem that it suffices to find aggregated proofs of satisfiability of these equations to get sublinear proofs in the number of wires circuit wires. For instance, a natural strategy would be to commit to wires with shrinking commitments and use any constant-size QA-NIZK argument of membership in linear spaces (e.g. [25]) to give an aggregated proof that the affine constraints hold and use "aggregated" variants of GS Proofs [18] such as [14,2] for the quadratic constraints.

The reason why this approach fails is that when using shrinking commitments it is unclear what are the guarantees provided by QA-NIZK arguments since they are not proofs of knowledge (w.r.t. general PPT adversaries and not generic ones). Similarly, the arguments for quadratic equations are commit-and-prove schemes which require binding commitments to the solution of the equation.

Knowledge Transfer Arguments. Our solution is to divide the set of constraints into d sets of quadratic and affine constraints, one per multiplicative level of the circuit. Namely, if $\phi : \mathbb{Z}_p^n \to \mathbb{Z}_p^{n'}$ is an arithmetic circuit of depth d, we express correct evaluation at level i as the following system:

- (quadratic constraints) $c_{ij} = a_{ij} b_{ij}$ for $j = 1, \dots, n_i$.
- (affine constraints) a_{ij}, b_{ij} are affine combinations of output wires of previous levels,

that is a_{ij}, b_{ij}, c_{ij} represent, respectively, the left, right and output of the jth gate at level i. Our technical innovation is to eliminate the need for binding commitments to the wires at all levels of the circuit by "transferring" knowledge of the input to lower levels.

More specifically, given adversarially chosen shrinking commitments L_i (resp. R_i, O_i) to all the left (resp. right, output) wires at level i, we first give a constant-size argument with constant-time verification which proves:

If $(\boldsymbol{a}_i, \boldsymbol{b}_i, L_i, R_i, O_i)$ is such that L_i, R_i open to $\boldsymbol{a}_i, \boldsymbol{b}_i$ then O_i opens to $\boldsymbol{c}_i = \boldsymbol{a}_i \circ \boldsymbol{b}_i$.

We think of this building block as a "quadratic knowledge transfer argument", as it shows that if an adversary knows an opening for left and right wires, it also knows an opening of the output wires at the next level. This property is formalized as a promise problem because the verifier of the argument never checks that L_i, R_i open to $\boldsymbol{a}_i, \boldsymbol{b}_i$ (otherwise the verification of the argument would be linear in the witness). Using a quadratic arithmetic program encoding [8] of the quadratic constraints we prove soundness under a certain q-assumption.

With this building block, the problem of constructing the argument is reduced to arguing that left and right wires are correctly assigned, i.e. proving that affine constraints are satisfied. We build a "linear knowledge transfer" argument with constant proof size and verification time showing that:

Given an opening of the commitments to the output wires $O_1, \ldots, O_i$ which is consistent with $L_1, \ldots, L_i$ and $R_1, \ldots, R_i$ then it is also consistent with L_{i+1} and R_{i+1}.

Correct evaluation of the circuit can be easily proven by combining these two building blocks. Since the input of the circuit is public and the shrinking commitments we use are deterministic, a consistent assignment $O_1, L_1, R_1, \ldots, O_d, L_d, R_d$ of the circuit wires is known by the reduction in the proof of soundness. A successful soundness adversary must output another assignment which disagrees with it starting from some level i. If the adversary outputs as part of its proof $L_1, \ldots, L_i, R_1, \ldots, R_i, O_1, \ldots, O_{i-1}, O_i^*$, with $O_i^* \neq O_i$, the reduction knows openings of L_i, R_i and it can break the soundness of the quadratic knowledge transfer argument. On the other hand, if it sends $L_1, \ldots, L_i^*, R_1, \ldots, R_i^*, O_1, \ldots, O_{i-1}$, where either $L_i^* \neq L_i$ or $R_i^* \neq R_i$, then it knows valid openings of O_j until level $i-1$ and it can break the soundness of the "linear knowledge transfer" argument.

To construct the linear knowledge transfer argument, we use QA-NIZK arguments of membership in linear spaces [20,21,27,25,14]. Although soundness of these arguments can be proven under standard assumptions, it turns out that traditional soundness is not what we need in this setting. Indeed, to see this, suppose we want to prove that two shrinking, deterministic commitments open to the same value. Let $\mathbf{M}, \mathbf{N}$ be the commitment keys. If $C_1 = \mathbf{M}\boldsymbol{w}$ and $C_2 = \mathbf{N}\boldsymbol{w}$ are commitments to the same value, obviously

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \in \mathbf{Im} \begin{pmatrix} \mathbf{M} \\ \mathbf{N} \end{pmatrix}. \tag{1}$$

Let π a QA-NIZK proof of membership in linear spaces for (1). In our linear knowledge transfer argument, π should convince the verifier that:

"If $C_1 = \mathbf{M}\boldsymbol{w}$ for some known $\boldsymbol{w}$, and π verifies, then $C_2 = \mathbf{N}\boldsymbol{w}$."

The problem is that for any $\boldsymbol{w}'$ such that $C_1 = \mathbf{M}\boldsymbol{w} = \mathbf{M}\boldsymbol{w}'$, an adversary can set $C_2 = \mathbf{N}\boldsymbol{w}'$ and compute π honestly with $\boldsymbol{w}'$. In other words, the adversary can "switch witnesses" without breaking the soundness of the QA-NIZK argument. So standard soundness does not help to argue that the left and right wires are consistently evaluated with lower levels of the circuit.

On the other hand, the "witness switching attack" is easy to rule out, as it requires the attacker to know two openings for C_1, but this breaks the binding property of the first commitment. However, because the commitment is shrinking we do not know how to extract $\boldsymbol{w}'$ to get a reduction to the binding property unless we use the knowledge soundness property of the QA-NIZK Argument as proven (in the generic group model) in [5].

Soundness of the Linear Argument under Standard Assumptions. One of our main technical contributions is to show that such witness switching attacks are not possible under a certain decisional assumption in bilinear groups. To get back to our example, our first observation is that, using the linear properties of the QA-NIZK arguments of membership in linear spaces, a break of the knowledge transfer property can be turned into a proof of membership $\pi^\dagger$ for a vector of the form $\left(\begin{smallmatrix} 0 \\ C \end{smallmatrix}\right)$, where $C = C_2 - \mathbf{N}\boldsymbol{w} \neq 0$.

The crs of the QA-NIZK argument system is of the form $\mathbf{A}, \mathbf{B} = \mathbf{M}^\top \mathbf{K}_1 + \mathbf{N}^\top \mathbf{K}_2, \mathbf{K}\mathbf{A}$, for some matrix $\mathbf{A}$ and a random matrices $\mathbf{K}_1, \mathbf{K}_2$. A proof for (C_1, C_2) must be of the form $C_1^\top \mathbf{K}_1 + C_2^\top \mathbf{K}_2$ (unless one solves some computationally hard problem). Intuitively, is not easy to construct $\pi^\dagger$ since it must be of the form $\pi^\dagger = C^\top \mathbf{K}_2$ and hence an adversary must somehow find an element in the kernel of $\mathbf{M}$ (which is in general a hard problem, otherwise the commitment is not binding) in order to eliminate any dependence on $\mathbf{K}_1$ in $\mathbf{B}$. However, in the security proof it is not clear how to extract such element in the kernel of $\mathbf{M}$, which is of the same size of $\boldsymbol{w}$, only from C and $\pi^\dagger$, which are of constant size. To bypass this problem, we assume that a stronger decisional assumption related to $\mathbf{M}$ holds, namely that it is hard to decide membership in the image of $\mathbf{M}^\top$ (a type of Matrix Diffie-Hellman assumption [4]). Specifically, we assume that $\mathbf{M}^\top \mathbf{K}_1$ is pseudo-random and, using this decisional assumption, we can jump to game where $\mathbf{K}_2$ is information theoretically hidden and then there is an exponentially low probability of computing $\pi^\dagger = C^\top \mathbf{K}_2$. To do this, we need to find a way around the problem that there is still some information about $\mathbf{K}_1$ which is leaked trough the crs of QA-NIZK arguments of [25] as $\mathbf{K}\mathbf{A} = \left(\begin{smallmatrix} \mathbf{K}_1\mathbf{A} \\ \mathbf{K}_2\mathbf{A} \end{smallmatrix}\right)$, where $\mathbf{A}$ is either a $(k+1) \times k$ matrix for general linear spaces or a $k \times k$ matrix when the linear spaces are generated by witness samplable distributions. To solve this, we use the fact that, information theoretically, part of $\mathbf{K}_1$ is never leaked through $\mathbf{K}\mathbf{A}$ when $\mathbf{A}$ is a $(k+1) \times k$ matrix. We leave it as an open question to achieve a similar result when $\mathbf{A}$ is a $k \times k$ to exploit witness samplability.

Zero-Knowledge. In all our subarguments the verification equations are pairing product equations, so they can be made zero-knowledge with Groth-Sahai proofs [18]. However, our proof uses in a fundamental way that the input of the verification is public. Therefore, this only works when the commitment to the input is extractable. The resulting scheme is not practical as this is only possible with bit-by-bit commitments to the input. However, it can be easily extended to boolean circuits with a proof size of $O(n - n_{pub} + n' + d)$ group elements (where n_{pub} is the size of the public input), which is an interesting improvement over state-of-the-art, as all constructions in the crs model under falsifiable assumptions are linear in the circuit size (see [17] and concrete improvements thereof, mainly [14]).

1.3 Previous Work

CRS NIZK for NP from Falsifiable Assumptions. Groth, Ostrovsky, and Sahai [17] constructed a NIZK proof system for boolean CircuitSat only from standard assumptions. Both the the size of the proof (in group elements) and the verifier's complexity (in group operations) depend asymptotically on the circuit size. The construction can be extended to arithmetic circuits using [18]. Several concrete improvements in the proof size can be done with recent results in the QA-NIZK setting [20,21,27,25,14] but we are not aware of any asymptotic improvements.

A trivial approach to reduce the proof size is to encrypt the witness using fully homomorphic encryption [9] and let the verifier evaluate the circuit homomorphically. Building on this idea, and using hybrid fully homomorphic encryption, Gentry et al. [10] constructed a proof of size $n + \mathsf{poly}(\lambda)$. While this shows that it is theoretically possible to build proofs of size independent of the circuit size under standard assumptions, they need to give NIZK proofs for correct key generation of FHE keys and correct evaluations of the FHE encryption algorithm and decryption algorithms.[4] These NIZK proofs, in general, need to represent the statements as boolean circuits and therefore they are of lower practical interest. Furthermore, note that the verifier needs to homomorphically evaluate the circuit using the FHE scheme, so its runtime is proportional to the circuit size.

A very recent result constructs proofs of size proportional to the circuit size plus an additive overhead in the security parameter (as opposed to multiplicative as in our work) in pairing based groups [24]. For NC1, one of the constructions is of size n (independent of the circuit size) plus an additive overhead in the security parameter. Although the verifier's runtime is proportional to the circuit size, it may be possible to preprocess the circuit dependent part and add it to the crs so that the verifier's runtime is only proportional to the size of the input. On the downside, the size of the crs is $O(n^3)$ as well as the underlying security assumption which is a q-assumption with q of size $O(n^3)$. Furthermore, the

[4] Note that using the celebrated recent results of Peikert and Shiehian [35] this scheme can be based solely on the LWE assumption.

additive overhead might be large as it hides a NIZK proof (computed with [17]) for the correct decryption of a ciphertex. Such a NIZK proof requires representing the decryption algorithm as a boolean circuit and to commit to each circuit wire.

Verifiable Computation. Kalai et al. [22], based on [13] and the sum-check protocol of Lund et al. [29], constructed the first publicly verifiable non-interactive delegation scheme for boolean circuits from a simple constant size assumption in bilinear groups. Their crs is circuit dependent but it can be universal using a crs for the universal circuit. [5]. The verifier's runtime is $O((n+d)\mathsf{polylog}(s))$, and the communication complexity is $O(d \cdot \mathsf{polylog}(\mathsf{s}))$ group elements, where s is the size of the circuit, and in most other parameters it is far from being efficient (crs size, prover complexity).

As explained in [22] there's a vast literature on verifiable computation (apart from the already mentioned) which can be roughly classified into a) designated verifier schemes [7,23], b) schemes under very strong assumptions: "knowledge of exponent" type (e.g. [8,34]), generic or algebraic group model (e.g.[16,30]), assumptions related to obfuscation, or homomorphic encryption [33] or c) interactive arguments [13]. Note that all these constructions are incomparable to ours as long as they either rely on arguably stronger assumptions (b) or are in a different model (a and c).

2 Preliminaries

Given some distribution $\mathcal{D}$ we denote by $x \leftarrow \mathcal{D}$ the process of sampling x according to $\mathcal{D}$. For a finite set S, $x \leftarrow S$ denotes an element sampled from the uniform distribution over S.

Bilinear Groups. Let $\mathcal{G}$ be some probabilistic polynomial time algorithm which on input 1^λ, where λ is the security parameter, returns the *group key* which is the description of an asymmetric bilinear group $gk = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \mathcal{P}_1, \mathcal{P}_2)$, where $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are groups of prime order p, the elements $\mathcal{P}_1, \mathcal{P}_2$ are generators of $\mathbb{G}_1, \mathbb{G}_2$ respectively, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is an efficiently computable, non-degenerate bilinear map, and there is no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

Elements in $\mathbb{G}_\gamma$, are denoted implicitly as $[a]_\gamma = a\mathcal{P}_\gamma$, where $\gamma \in \{1, 2, T\}$ and $\mathcal{P}_T = e(\mathcal{P}_1, \mathcal{P}_2)$. With this notation, $e([a]_1, [b]_2) = [ab]_T$. Vectors and matrices are denoted in boldface. Given a matrix $\mathbf{T} = (t_{i,j})$, $[\mathbf{T}]_\gamma$ is the natural embedding of $\mathbf{T}$ in $\mathbb{G}_\gamma$, that is, the matrix whose (i, j)th entry is $t_{i,j}\mathcal{P}_\gamma$. We use the notation (a, b) to refer to a elements of $\mathbb{G}_1$ and b elements of $\mathbb{G}_2$.

$\mathbf{I}_n$ refers to the identity matrix in $\mathbb{Z}_p^{n\times n}$, $\mathbf{0}_{m\times n}$ to the all-zero matrix in $\mathbb{Z}_p^{m\times n}$ (simply $\mathbf{I}$ and $\mathbf{0}$, respectively, if n and m are clear from the context).

[5] There's the technicality that a verifier running in time sub-linear in the circuit size can not even read the circuit, which is part of the input of the universal circuit. For this reason, they restricted the circuits to be log space uniform boolean cicuits

Lagrangian Pedersen Commitments. Given an arbitrary set $\mathcal{R} = \{r_1, \ldots, r_m\} \subset \mathbb{Z}_p$, we define the jth Lagrange interpolation polynomial as:

$$\lambda_j(X) = \prod_{\ell \neq j} \frac{(X - r_\ell)}{(r_j - r_\ell)}.$$

It is a well known fact that given a set of values x_j, $j = 1, \ldots, m$, $P(X) = \sum_{j=1}^{m} x_j \lambda_j(X)$ is the unique polynomial of degree at most $m - 1$ such that $P(r_j) = x_j$. The Lagrangian Pedersen commitment in $\mathbb{G}_\gamma$ for some $\gamma \in \{1, 2\}$ to a vector $x \in \mathbb{Z}_p^m$ is defined as

$$\mathsf{Com}_{ck}(\boldsymbol{x}) = \sum_{i=1}^{m} x_j [\lambda_j(s)]_\gamma = [P(s)]_\gamma,$$

where the commitment key is $ck = ([\lambda_1(s)]_\gamma, \ldots, [\lambda_m(s)]_\gamma)$, for $s \leftarrow \mathbb{Z}_p$. It is computationally binding under the m-DLog assumption.

We also consider vectors of Lagrangian Pedersen commitments defined as $[P(\boldsymbol{s})]_\gamma = \sum_{i=1}^{m} x_i [\lambda_i(\boldsymbol{s})]_\gamma \in \mathbb{G}_\gamma^{k_s}$, where $\boldsymbol{s} \in \mathbb{Z}_p^{k_s}$ for some $k_s \in \mathbb{N}$ and $\lambda_i(\boldsymbol{s})$ is just $(\lambda_i(s_1), \ldots, \lambda_i(s_{k_s}))^\top$.

2.1 Cryptographic Assumptions

Definition 1. *Let $k \in \mathbb{N}$. We call $\mathcal{D}_{\ell,k}$ (resp. $\mathcal{D}_k$) a matrix distribution if it outputs in PPT time, with overwhelming probability matrices in $\mathbb{Z}_p^{\ell \times k}$ (resp. in $\mathbb{Z}_p^{(k+1) \times k}$). For a matrix distribution $\mathcal{D}_k$, we denote as $\overline{\mathcal{D}}_k$ the distribution of the first k rows of the matrices sampled according to $\mathcal{D}_k$.*

Assumption 1 *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution and $gk \leftarrow \mathcal{G}(1^\lambda)$. For all non-uniform PPT adversaries $\mathcal{A}$ and relative to $gk \leftarrow \mathcal{G}(1^\lambda)$, $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}, \boldsymbol{w} \leftarrow \mathbb{Z}_p^k, [\boldsymbol{z}]_\gamma \leftarrow \mathbb{G}_\gamma^\ell$ and the coin tosses of adversary $\mathcal{A}$,*

1. *the Matrix Decisional Diffie-Hellman Assumption in $\mathbb{G}_\gamma$ ($\mathcal{D}_k$-MDDH_γ) holds if*

$$|\Pr[\mathcal{A}(gk, [\mathbf{A}]_\gamma, [\mathbf{A}\boldsymbol{w}]_\gamma) = 1] - \Pr[\mathcal{A}(gk, [\mathbf{A}]_\gamma, [\boldsymbol{z}]_\gamma) = 1]| \leq \mathsf{negl}(\lambda),$$

2. *the Split Matrix Decisional Diffie-Hellman Assumption in $\mathbb{G}_\gamma$ ($\mathcal{D}_k$-SMDDH_γ) holds if*

$$|\Pr[\mathcal{A}(gk, [\mathbf{A}]_1, [\mathbf{A}]_2, [\mathbf{A}\boldsymbol{w}]_\gamma) = 1] - \Pr[\mathcal{A}(gk, [\mathbf{A}]_1, [\mathbf{A}]_2, [\boldsymbol{z}]_\gamma) = 1]| \leq \mathsf{negl}(\lambda).$$

Two examples of interesting distributions are the following:

$$\mathcal{L}_k : \mathbf{A} = \begin{pmatrix} s_1 & 0 & \ldots & 0 \\ 0 & s_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & s_k \\ 1 & 1 & \ldots & 1 \end{pmatrix} \qquad \mathcal{LG}_{\mathcal{R},k} : \mathbf{A} = \begin{pmatrix} \lambda_1^{\mathcal{R}}(s_1) & \lambda_1^{\mathcal{R}}(s_2) & \ldots & \lambda_1^{\mathcal{R}}(s_k) \\ \lambda_2^{\mathcal{R}}(s_1) & \lambda_2^{\mathcal{R}}(s_2) & \ldots & \lambda_2^{\mathcal{R}}(s_k) \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_\ell^{\mathcal{R}}(s_1) & \lambda_\ell^{\mathcal{R}}(s_2) & \ldots & \lambda_\ell^{\mathcal{R}}(s_k) \end{pmatrix},$$

where $s_i \leftarrow \mathbb{Z}_p$ and $\mathcal{R} = \{r_1, \ldots, r_N\} \subset \mathbb{Z}_p$. The assumption associated to the first distribution is the k-Lin family. The assumption associated to the second one is new to this paper and is the $(\mathcal{R}, k)$-Lagrangian Assumption. In our construction, we will use the $\mathcal{LG}_{\mathcal{R},2}$-$\mathsf{SMDDH}_1$ assumption (for N the maximum number of gates of the same multiplicative depth). In the full version of this work we argue about the generic hardness of the $\mathcal{LG}_{\mathcal{R},2}$-$\mathsf{MDDH}_\gamma$ assumption in symmetric bilinear groups, which implies the generic hardness of $\mathcal{LG}_{\mathcal{R},2}$-$\mathsf{SMDDH}_1$ in asymmetric bilinear groups.

We note that for all interesting distributions $\mathcal{D}_k$, we can assume that the $\mathcal{D}_k$-MDDH Assumption is generically hard in k-linear groups and in particular, that every $k \times k$ minor is invertible with overwhelming probability.

The Kernel Diffie-Hellman Assumption [31] says one cannot find a non-zero vector in one of the groups which is in the co-kernel of $\mathbf{A}$. We also use a generalization in bilinear groups which says one cannot find a pair of vectors in $\mathbb{G}_1^{k+1} \times \mathbb{G}_2^{k+1}$ such that the difference of the vector of their discrete logarithms is in the co-kernel of $\mathbf{A}$.

Assumption 2 *Let $\mathcal{D}_{\ell,k}$ be a matrix distribution. For all non-uniform PPT adversaries $\mathcal{A}$ and relative to $gk \leftarrow \mathcal{G}(1^\lambda)$, $\mathbf{A} \leftarrow \mathcal{D}_{\ell,k}, \boldsymbol{w} \leftarrow \mathbb{Z}_p^k, [\boldsymbol{z}]_\gamma \leftarrow \mathbb{G}_\gamma^\ell$ and the coin tosses of adversary $\mathcal{A}$,*

1. *the Find-Rep Assumption holds if*
$$\Pr\left[\boldsymbol{r} \leftarrow \mathcal{A}(gk, [\mathbf{A}]_1, [\mathbf{A}]_2) : \boldsymbol{r}^T\mathbf{A} = \mathbf{0}\right] = \mathsf{negl}(\lambda),$$
2. *the Kernel Matrix Diffie-Hellman Assumption holds in $\mathbb{G}_\gamma$ [31] if*
$$\Pr\left[[\boldsymbol{r}]_{3-\gamma} \leftarrow \mathcal{A}(gk, [\mathbf{A}]_\gamma) : \boldsymbol{r}^\top\mathbf{A} = 0\right] = \mathsf{negl}(\lambda),$$
3. *the Split Kernel Matrix Diffie-Hellman Assumption [14] holds if*
$$\Pr\left[[\boldsymbol{r}]_1, [\boldsymbol{s}]_2 \leftarrow \mathcal{A}(gk, [\mathbf{A}]_1, [\mathbf{A}]_2) : \boldsymbol{r} \neq \boldsymbol{s} \wedge \boldsymbol{r}^\top\mathbf{A} = \boldsymbol{s}^\top\mathbf{A}\right] = \mathsf{negl}(\lambda).$$

The Find-Rep Assumption for the $\mathcal{LG}_{\mathcal{R},\ell,k}$ MDH Assumption is equivalent to solving k instances of the q-Dlog Assumption in both groups, in which the adversary receives q powers of s_i, $i = 1, \ldots, k$ in both groups and computes $s_i \in \mathbb{Z}_p$. This follows from the observation that if $\boldsymbol{r}$ is a solution of the Find-Rep problem, it can be associated to a polynomial which is 0 in s_i for all $i = 1, \ldots, k$ and its factorization allows to compute s_i.

We note that the Split Decisional and Split Kernel MDH Assumptions are generically hard in asymmetric bilinear groups for all distributions for which the non split variant is hard in symmetric bilinear groups whenever $k \geq 2$.

Finally, we introduce an assumption which is similar to the q-SFrac Assumption considered in [12], but in the source group.

Assumption 3 ($\mathcal{R}$-RSDH Assumption) *Let $\mathcal{R}$ be an arbitrary set of integers of cardinal q. The $\mathcal{R}$-Rational Strong Diffie-Hellman Assumption holds in $\mathbb{G}_1$ if the following probability is negligible in λ:*

$$\Pr\left[\begin{array}{c|c} e([z]_1, [1]_2) = e([w]_1, [t(s)]_2) & gk \leftarrow \mathcal{G}(1^\lambda); \\ z \neq 0 & ([z]_1, [w]_1) \leftarrow \mathcal{A}\left(gk, \mathcal{R}, \left\{[s^i]_{1,2}\right\}_{i=1}^{q-1}, [s^q]_2\right) \end{array}\right],$$

where $t(s) = \prod_{r \in \mathcal{R}}(s - r)$, and the probability is taken over $gk \leftarrow \mathcal{G}(1^\lambda)$, $s \leftarrow \mathbb{Z}_p$ and the coin tosses of adversary $\mathcal{A}$.

It is important to note that it is possible to check if an adversary has succeeded in breaking the assumption, since the value $[t(s)]_2$ can be constructed as a linear combination of $\{[s^i]_2\}_{i=1}^q$ given $\mathcal{R}$.

The intuition why the assumption is generically hard is as follows. Since $[z]_1, [w]_1$ are given in the group $\mathbb{G}_1$, the adversary must construct them as a linear combinations of all elements it has received in $\mathbb{G}_1$, which are $([1]_1, [s]_1, \ldots, [s^{q-1}]_1)$. On the other hand, the adversary can only win if $z/t(s) = w$, but the adversary can only find a non-trivial solution generically if z is constructed as a (non-zero) multiple of $t(X) = \prod_{r \in \mathcal{R}}(X - r)$ evaluated at s. But this is not possible because in $\mathbb{G}_1$ it only receives powers of s of degree at most $q - 1$ and $t(X)$ is of degree q.

3 Arithmetic Circuits

Arithmetic circuits are acyclic directed graphs where the edges are called wires and the vertices are called gates. Gates with in-degree 0 are labeled by variables X_i, $i = 1, \ldots, n$ or with a constant field element, the rest of the gates are either labeled with $\times$ and are referred to as multiplication gates or with $+$ and are called addition gates. In this work we consider only fan-in 2 multiplication gates and the circuit is defined over a field $\mathbb{Z}_p$, where p is the order of some cryptographically useful bilinear group. Each circuit computes a function $\phi : \mathbb{Z}_p^n \rightarrow \mathbb{Z}_p^{n'}$.

Let $\mathcal{G}$ be the set of multiplicative gates of the circuit excluding multiplication-by-constant gates. We denote by m the cardinal of this set. For simplicity and without loss of generality, we may assume all outputs of the circuit to be the output of some multiplication gate.

For our construction of Sect. 5, we partition the set $\mathcal{G}$ of multiplicative gates of the circuit into different levels. More precisely, we define $\{\mathcal{G}_i\}_{i=1}^{d'}$, where $\mathcal{G}_i$, for $i = 1, \ldots, d'$, is the set of gates $G \in \mathcal{G}$ such that the maximum of gates in $\mathcal{G}$ evaluated in any path from the input of the circuit to an input of G is $i - 1$. The minimal such d' for which the partition exists is the multiplicative depth of the circuit, which we always denote by d. Further, we define $\mathcal{G}_0$ to be the set of n_0 variable inputs. If $G \in \mathcal{G}_i$, we say that G has multiplicative depth i. Let n_i be the cardinal of $\mathcal{G}_i$. With this notation, a circuit computes a function $\phi : \mathbb{Z}_p^{n_0} \rightarrow \mathbb{Z}_p^{n_d}$, i.e. $n = n_0$, $n' = n_d$ and the number of multiplication gates is $\sum_{i=1}^d n_i$.

We now consider an encoding of circuit satisfiability where the variables are divided according to their multiplicative depth. For each gate in $\mathcal{G}_i$, $i \in \{1, \ldots, d\}$ the circuit is correctly evaluated if the output of the gate is the product of two multivariate polynomials of degree 1 where the variables are outputs of gates of less multiplicative depth, that is, the output of gates in $\mathcal{G}_j$, for some j, $0 \leq j \leq i - 1$.

Lemma 1. *Let $\phi : \mathbb{Z}_p^{n_0} \rightarrow \mathbb{Z}_p^{n_d}$, be a circuit of multiplicative depth d and with m gates. For $i \in \{1, \ldots, d\}$, define n_i as the number multiplication gates at level i. There exist*

a) variables C_{ij}, $i = 0, \dots, d$, $j = 1, \dots, n_i$,
b) variables A_{ij}, B_{ij}, $i = 1, \dots, d$, $j = 1, \dots, n_i$,
b) constants $f_{ij}, g_{ij}, f_{ijk\ell}, g_{ijk\ell} \in \mathbb{Z}_p$, $i = 1, \dots, d$, $k = 0, \dots, i-1$, $j = 1, \dots, n_i$, $\ell = 1, \dots, n_k$

such that, for every $(x_1, \dots, x_{n_0}) \in \mathbb{Z}_p^{n_0}$, if we set $C_{0j} = x_j$, for all $j = 1, \dots, n_0$, then $\phi(x_1, \dots, x_{n_0}) = (y_1, \dots, y_{n_d})$ and for each $i \in \{1, \dots, d\}$, A_{ij}, B_{ij}, C_{ij} are evaluated respectively to the left, the right and the output wires of the jth gate at level i, if and only if the following equations are satisfied:

1. *(Quadratic Constraints). For each $i = 1, \dots, d$, if $j = 1, \dots, n_i$: $C_{ij} = A_{ij} B_{ij}$.*
2. *(Affine Constraints) $A_{ij} = f_{ij} + \sum_{k=0}^{i-1} \sum_{\ell=1}^{n_k} f_{ijk\ell} C_{k\ell}$ and $B_{ij} = g_{ij} + \sum_{k=0}^{i-1} \sum_{\ell=1}^{n_k} g_{ijk\ell} C_{k\ell}$.*
3. *(Correct Output) $C_{dj} = y_j$, $j = 1, \dots, n_d$.*

Given an arithmetic circuit $\phi : \mathbb{Z}_p^{n_0} \to \mathbb{Z}_p^{n_d}$, we can define the witness for correct evaluation of $\phi(\boldsymbol{x}) = \boldsymbol{y}$ as a tuple $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$, where $\boldsymbol{a} = (\boldsymbol{a}_1, \dots, \boldsymbol{a}_d)$, $\boldsymbol{b} = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_d)$, $\boldsymbol{c} = (\boldsymbol{c}_0, \dots, \boldsymbol{c}_d)$, $\boldsymbol{s}_i = (s_{i1}, \dots, s_{in_i})$ for any $s \in \{a, b, c\}$. The tuple is an an assignment to A_{ij}, B_{ij} and C_{ij} which satisfies the equations described in Lemma 1.

Using standard techniques due to [8], quadratic constraints can be written as a polynomial divisibility problem.

Lemma 2. *(QAP for the Hadamard Product) Let $(\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i) \in (\mathbb{Z}_p^{n_i})^3$, $n_i \in \mathbb{N}$. Let $\mathcal{R} = \{r_1, \dots, r_N\} \subset \mathbb{Z}_p$ be a set of elements of $\mathbb{Z}_p$ for some $N \geq n_i$ and let $\lambda_i(X) = \prod_{j \neq i} \dfrac{X - r_j}{r_i - r_j}$. Define*

$$p_i(X) = \left(\sum_{j=1}^{n_i} a_{ij} \lambda_j(X) \right) \left(\sum_{j=1}^{n_i} b_{ij} \lambda_j(X) \right) - \left(\sum_{j=1}^{n_i} c_{ij} \lambda_j(X) \right).$$

Then, $\boldsymbol{c}_i = \boldsymbol{a}_i \circ \boldsymbol{b}_i$ if and only if $p_i(X) = h_i(X) t(X)$, where $t(X) = \prod_{r \in \mathcal{R}} (X - r)$ and $h_i(X) \in \mathbb{Z}_p[X]$ is a polynomial of degree at most $N - 2$.

Proof. By definition, $p_i(r_j) = a_{ij} b_{ij} - c_{ij}$, so $p_i(X)$ is divisible by $t(X)$ if and only if $a_{ij} b_{ij} - c_{ij} = 0$ for all $j = 1, \dots, n_i$.

On the other hand, for each i, affine constraints can be written also as polynomial relations. That is, for any set $\mathcal{R} = \{r_1, \dots, r_N\}$ such that $N \geq n_i$, there exist families of polynomials $\mathcal{V} = \{v_i, v_{ik\ell}\}$, $\mathcal{W} = \{w_i, w_{ik\ell}\}$ of degree $N - 1$ such that $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ is a valid witness if and only if $\sum_{j=1}^{n_i} a_{ij} \lambda_j(X) = v_i(X) + \sum_{k=0}^{i-1} \sum_{\ell=1}^{n_k} c_{kl} v_{ik\ell}(X)$ and $\sum_{j=1}^{n_i} b_{ij} \lambda_j(X) = w_i(X) + \sum_{k=0}^{i-1} \sum_{\ell=1}^{n_k} c_{kl} w_{ik\ell}(X)$. It suffices to define $v_i(X) = \sum_{j=1}^{n_i} f_{ij} \lambda_j(X)$, $v_{ik\ell}(X) = \sum_{j=1}^{n_i} f_{ijk\ell} \lambda_j(X)$, $w_i(X) = \sum_{j=1}^{n_i} g_{ij} \lambda_j(X)$, $w_{ik\ell}(X) = \sum_{j=1}^{n_i} g_{ijk\ell} \lambda_j(X)$. The proof follows by evaluating the equations in the points $r_j \in \mathcal{R}$.

4 Arguments of Knowledge Transfer

In this section we construct what we informally name "knowledge transfer argument" for both linear and quadratic equations. The name captures the idea that these arguments ensure that if a valid opening is known for some committed value, then an opening is also known for another commitment and this second opening is a certain quadratic or linear function of the original opening.

Formally, the prover needs to prove membership in a language $\mathcal{L}$ of the form $(\boldsymbol{w}, C, D)$, where $\boldsymbol{w}$ is the opening of a shrinking commitment C. The statement is that "if C opens to $\boldsymbol{w}$, then D opens to $F(\boldsymbol{w})$". Since typically there is an exponential number of possible openings of C, the language would not make sense without $\boldsymbol{w}$, i.e. the statement "there exists an opening $\boldsymbol{w}$ of C such that $\boldsymbol{D}$ opens to $F(\boldsymbol{w})$" would most probably be always true.

Deciding membership in $\mathcal{L}$ can be done efficiently with a number of operations which is proportional to the size of the statement. Our verifier, however, does not use $\boldsymbol{w}$ for verification (i.e. it never checks that $\boldsymbol{w}$ is a valid opening of C) and does only a constant number of public key operations (ignoring the need to read $\boldsymbol{w}$ as part of the statement). When using these subarguments in the full argument for correct circuit evaluation, the verifier never reads $\boldsymbol{w}$ but $\boldsymbol{w}$ is uniquely determined by the context.

This is formalized as a promise problem defined by a language of good instances $\mathcal{L}_{YES}$ and of bad instances $\mathcal{L}_{NO}$. Completeness guarantees that proofs are accepted for all instances of $\mathcal{L}_{YES}$, while soundness guarantees that no argument will be accepted for instances of $\mathcal{L}_{NO}$. The promise is that "$\boldsymbol{w}$ is an opening of C" and nothing is claimed when $x \notin (\mathcal{L}_{YES} \cup \mathcal{L}_{NO})$ (i.e. when the promise does not hold). A formal definition of QA arguments for promise problems can be found in the full version of this work.

4.1 Argument for Hadamard Products

Let $m \in \mathbb{N}$. We give an argument for the promise problem defined by languages $\mathcal{L}^{\mathsf{quad}}_{YES}, \mathcal{L}^{\mathsf{quad}}_{NO}$, which are parameterized by $m \in \mathbb{N}$ and a Lagrangian Pedersen commitment key $ck = ([\boldsymbol{\Lambda}]_1, [\boldsymbol{\Lambda}]_2)$ and are defined as

$$\mathcal{L}^{\mathsf{quad}}_{YES} = \left\{ \begin{array}{l} (\boldsymbol{a}, \boldsymbol{b}, [L]_1, [R]_2, [O]_1) : \boldsymbol{c} = \boldsymbol{a} \circ \boldsymbol{b} \\ \text{ and } [L]_1 = [\boldsymbol{\Lambda}]_1 \boldsymbol{a}, [R]_2 = [\boldsymbol{\Lambda}]_2 \boldsymbol{b}, [O]_1 = [\boldsymbol{\Lambda}]_1 \boldsymbol{c} \end{array} \right\},$$
$$\mathcal{L}^{\mathsf{quad}}_{NO} = \left\{ \begin{array}{l} (\boldsymbol{a}, \boldsymbol{b}, [L]_1, [R]_2, [O]_1) : \boldsymbol{c} = \boldsymbol{a} \circ \boldsymbol{b}, \\ [L]_1 = [\boldsymbol{\Lambda}]_1 \boldsymbol{a} \text{ and } [R]_2 = [\boldsymbol{\Lambda}]_2 \boldsymbol{b}, \\ \text{ but } [O]_1 \neq [\boldsymbol{\Lambda}]_1 \boldsymbol{c} \end{array} \right\}.$$

Perfect completeness. The argument described in Fig. 1 has perfect completeness as the values $[L]_1, [O]_1$ can be computed from $\{[\lambda_i(s)]_1 \ldots, [\lambda_m(s)]_1\}$, and $[R]_2$ from $\{[\lambda_i(s)]_2 \ldots, [\lambda_m(s)]_2\}$. Further, by definition, the polynomial $\ell(X)r(X) - o(X)$ takes the value $a_i b_i - c_i = 0$ at point $r_i \in \mathcal{R}$. Therefore,

```
$\underline{\mathsf{K}(gk, \mathcal{R})}$:
    Sample $s \leftarrow \mathbb{Z}_p^*$;
    Output crs =
    $\Big(gk, \{[\lambda_1(s)]_\gamma, \ldots, [\lambda_m(s)]_\gamma\}_{\gamma \in \{1,2\}},$
    $\{[s^i]_1\}_{i \in \{1,\ldots,m-2\}}, [t(s)]_2\Big)$.

$\underline{\mathsf{V}(\mathrm{crs}, \boldsymbol{a}, \boldsymbol{b}, [L]_1, [R]_2, [O]_1, [H]_1)}$:
    Check if:
    $e([L]_1, [R]_2) - e([O]_1, [1]_2) = e([H]_1, [t(s)]_2)$;
    output 1 in this case and 0 otherwise.

$\underline{\mathsf{P}(\mathrm{crs}, \boldsymbol{a}, \boldsymbol{b})}$:
    $\ell(X) = \sum_{i=1}^m a_i \lambda_i(X)$;
    $r(X) = \sum_{i=1}^m b_i \lambda_i(X)$;
    $o(X) = \sum_{i=1}^m c_i \lambda_i(X)$;
    $h(X) = (\ell(X) r(X) - o(X))/t(X)$;
    $[L]_1 = [\ell(s)]_1$; $[R]_2 = [r(s)]_2$;
    $[O]_1 = [o(s)]_1$; $[H]_1 = [h(s)]_1$;
    Output $[H]_1$.
```

Fig. 1. Our argument for Hadamard products. $\lambda_i(X)$ is the ith Lagrange polynomial associated to $\mathcal{R}$, a set of $\mathbb{Z}_p$ of cardinal m, $t(X)$ is the polynomial which has as roots all the elements of $\mathcal{R}$. Both $\boldsymbol{a}$ and $\boldsymbol{b}$ are m-dimensional vectors in $\mathbb{Z}_p$.

$\ell(X)r(X) - o(X)$ is divisible by $t(X)$, so $h(X)$ is well defined. Further, the degree of H is at most $m-2$ (since $\ell(X)r(X)$ has degree $2m-2$ and $t(X)$ has degree m) and thus $[H]_1$ can be computed from $\{[s]_1, \ldots, [s^{m-2}]_1\}$.

Computational Soundness. We argue that if $\mathcal{A}$ produces an accepting proof for $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, [L]_1, [R]_2, [O]_1) \in \mathcal{L}_{NO}^{\mathsf{quad}}$ then we can construct an adversary $\mathcal{B}$ against the $(\mathcal{R}, \boldsymbol{m})$-Rational Strong Diffie-Hellman Assumption. Given a challenge gk, $\{[s^i]_1\}_{i=1}^{m-1}, \{[s^i]_2\}_{i=1}^{m}$, adversary $\mathcal{B}$ can simulate the common reference string perfectly because $\lambda_i(X)$ is a polynomial whose coefficients in $\mathbb{Z}_p$ depend only on $\mathcal{R}$ of degree at most $m-1$. Therefore, $[\lambda_i(s)]_1, [\lambda_i(s)]_2$ can be computed from $\{s^i\}_{i=1}^{m-1}$ in both the source groups. On the other hand, $t(X)$ is a polynomial with coefficients in $\mathbb{Z}_p$ which depend only on $\mathcal{R}$ of degree at most m. So $[t(s)]_2$ can be computed in $\mathbb{G}_2$ given $\{[s^i]_2\}_{i=1}^{m}$.

Adversary $\mathcal{A}$ outputs $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, [L]_1, [R]_2, [O^\dagger]_1, [H^\dagger]_1)$ which is accepted by the verifier and $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, [L]_1, [R]_2, [O^\dagger]_1) \in \mathcal{L}_{NO}^{\mathsf{quad}}$, which in particular means that, for $L = \ell(s)$, $R = r(s)$, the equation

$$e([L]_1, [R]_2) - e([O^\dagger]_1, [1]_2) = e([H^\dagger]_1, [t(s)]_2) \qquad (2)$$

holds but $O^\dagger \neq O(s)$.

Since adversary $\mathcal{B}$ received $\boldsymbol{a}, \boldsymbol{b}$ as part of $\mathcal{A}$'s output, it can run the honest prover algorithm and obtain O, H which satisfy that

$$e([L]_1, [R]_2) - e([O]_1, [1]_2) = e([H]_1, [t(s)]_2) \qquad (3)$$

and $O = O(s)$.

Subtracting equations (2) and (3), we get $e([O^\dagger - O]_1, [1]_2) = e([H^\dagger - H]_1, [t(s)]_2)$. Therefore, $([O^\dagger - O]_1, [H^\dagger - H]_1)$ is a solution to the $(\mathcal{R}, \boldsymbol{m})$-Rational Strong Diffie-Hellman Assumption.

We note that the verification algorithm never uses $(\boldsymbol{a}, \boldsymbol{b})$ which are part of the statement. When using the scheme as a building block, we omit $(\boldsymbol{a}, \boldsymbol{b})$ from the input of the verifier of the quadratic relations.

4.2 Argument for Linear Languages

Let gk be a bilinear group of order p and $\ell_1, \ell_2, n \in \mathbb{N}$ and $[\mathbf{M}]_1 \in \mathbb{G}_1^{\ell_1 \times n}, [\mathbf{N}]_1 \in \mathbb{G}_1^{\ell_2 \times n}$ be some matrices sampled from some distributions $\mathcal{M}, \mathcal{N}$. We give two different arguments for the promise problem defined by languages $\mathcal{L}_{YES}^{\mathsf{lin}}, \mathcal{L}_{NO}^{\mathsf{lin}}$, which are parameterized by $gk, [\mathbf{M}]_1, [\mathbf{N}]_1$ and are defined as:

$$\mathcal{L}_{YES}^{\mathsf{lin}} = \{(\boldsymbol{w}, [\boldsymbol{u}]_1, [\boldsymbol{v}]_1) : [\boldsymbol{u}]_1 = [\mathbf{M}]_1 \boldsymbol{w},\ [\boldsymbol{v}]_1 = [\mathbf{N}]_1 \boldsymbol{w}\}$$
$$\mathcal{L}_{NO}^{\mathsf{lin}} = \{(\boldsymbol{w}, [\boldsymbol{u}]_1, [\boldsymbol{v}]_1) : [\boldsymbol{u}]_1 = [\mathbf{M}]_1 \boldsymbol{w},\ [\boldsymbol{v}]_1 \neq [\mathbf{N}]_1 \boldsymbol{w}\}.$$

The arguments are simply the QA-NIZK Arguments of membership in linear spaces for general and witness samplable distributions as presented by Kiltz and Wee [25] (which generalize previous constructions [26,21]). Both arguments are very similar and can be easily written in a unified way. The idea is to use the arguments to prove that there exists a witness $\boldsymbol{w}$ such that $\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} = \begin{pmatrix} \mathbf{M} \\ \mathbf{N} \end{pmatrix} \boldsymbol{w}$. Intuitively, assuming that it is hard to find non-trivial $(\boldsymbol{w}, \boldsymbol{w}')$ such that $[\boldsymbol{u}]_1 = [\mathbf{M}]_1 \boldsymbol{w} = [\mathbf{M}]_1 \boldsymbol{w}'$, this would prove that $[\boldsymbol{v}]_1 = [\mathbf{N}]_1 \boldsymbol{w}$. However, finding a security proof is not simple.

For witness samplable distributions, we only know a proof in the generic group model. The proof is a trivial consequence of the knowledge soundness property of QA-NIZK arguments which has already been used in previous works [5]. It has a proof size of k group elements when instantiated for the k-Lin Assumption.

Our main technical contribution is to prove soundness for the promise problem for general distributions (not necessarily witness samplable) assuming the hardness of the decisional problem for the distribution associated to matrix $\mathbf{M}$ (the $\mathcal{M}^\top$-MDDH Assumption). It has a proof size of $k+1$ group elements when instantiated for the k-Lin Assumption.

In Fig. (2) we describe the QA-NIZK argument of membership in linear spaces for witness samplable and general distributions (the only difference between these two cases is the definition of $\tilde{\mathcal{D}}_k$), as presented in [25]. The difference with the original presentation in [25] is that we separate the key $\mathbf{K}$ in blocks $\mathbf{K}_1, \mathbf{K}_2$ associated to $\mathbf{M}, \mathbf{N}$, which will be convenient for the proof. Perfect completeness, perfect zero-knowledge and computational soundness under any $\mathcal{D}_k$-KerMDH Assumption is proven [25].

Soundness of $\mathsf{Lin}_{\tilde{\mathcal{D}}_k}$, w.r.t. the language $\mathcal{L}_{NO}^{\mathsf{lin}}$, is a direct consequence of Lemma 3.

Lemma 3. *For any adversary $\mathcal{A}$ and for any $\mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}$, let*

$$\epsilon_{\mathcal{A}} = \Pr\left[\begin{array}{c} \boldsymbol{v} \neq 0 \\ \boldsymbol{\pi} = \boldsymbol{v}^\top \mathbf{K}_2 \end{array} \middle| \begin{array}{l} \mathbf{M} \leftarrow \mathcal{M}; \mathbf{N} \leftarrow \mathcal{N}; \\ \mathrm{crs} \leftarrow \mathsf{K}(gk, [\mathbf{M}]_1, [\mathbf{N}]_1); \\ ([\boldsymbol{v}]_1, [\boldsymbol{\pi}]_1) \leftarrow \mathcal{A}(\mathrm{crs}, [\mathbf{M}]_1, [\mathbf{N}]_1) \end{array}\right].$$

```
$\underline{\mathsf{K}(gk, [\mathbf{M}]_1, [\mathbf{N}]_1)}$: // $\mathbf{M} \in \mathbb{Z}_p^{\ell_1 \times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}$
    $\mathbf{K}_1 \leftarrow \mathbb{Z}_p^{\ell_1 \times \overline{k}}$; $\mathbf{K}_2 \leftarrow \mathbb{Z}_p^{\ell_2 \times \overline{k}}$;
    $\mathbf{K} = \begin{pmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \end{pmatrix}$;
    Sample $\mathbf{A} \leftarrow \widetilde{\mathcal{D}}_k$;
    $[\mathbf{B}]_1 = [\mathbf{M}^\top \mathbf{K}_1 + \mathbf{N}^\top \mathbf{K}_2]_1$;
    $\mathbf{C}_1 = \mathbf{K}_1 \mathbf{A}$; $\mathbf{C}_2 = \mathbf{K}_2 \mathbf{A}$; $\mathbf{C} = \mathbf{K}\mathbf{A}$
    return crs $= (gk, [\mathbf{B}]_1, [\mathbf{A}]_2, [\mathbf{C}]_2)$.

$\underline{\mathsf{P}(\mathrm{crs}, [\boldsymbol{u}]_1, [\boldsymbol{v}]_1, \boldsymbol{w})}$:
    return $[\boldsymbol{\pi}]_1 = \boldsymbol{w}^\top [\mathbf{B}]_1$;

$\underline{\mathsf{V}(\mathrm{crs}, [\boldsymbol{u}]_1, [\boldsymbol{v}]_1, [\boldsymbol{\pi}]_1)}$:
    Check if:
    $e([\boldsymbol{\pi}]_1, [\mathbf{A}]_2) =$
    $e([\boldsymbol{u}^\top]_1, [\mathbf{C}_1]_2) + e([\boldsymbol{v}^\top]_1, [\mathbf{C}_2]_2)$
```

Fig. 2. The $\mathsf{Lin}_{\widetilde{\mathcal{D}}_k}$ argument for proving membership in linear spaces. The matrix $\mathbf{A}$ is either sampled from a distribution $\widetilde{\mathcal{D}}_k = \overline{\mathcal{D}}_k$ or from a distribution $\widetilde{\mathcal{D}}_k = \mathcal{D}_k$, such that the $\mathcal{D}_k$-KerMDH assumption holds. In the latter case $\overline{k} = k + 1$ while in the former case $\overline{k} = k$.

1. *When $\widetilde{\mathcal{D}}_k = \overline{\mathcal{D}}_k$ and $\mathcal{M}$ is witness samplable, if $\mathcal{A}$ is generic there exists a PPT adversary $\mathcal{B}$ such that $\epsilon_{\mathcal{A}} \leq \mathsf{Adv}_{\mathcal{M}\text{-}\mathsf{FindRep}}(\mathcal{B}) + \mathsf{negl}(\lambda)$.*
2. *When $\widetilde{\mathcal{D}}_k = \mathcal{D}_k$, there exists a PPT adversary $\mathcal{B}$ such that $\epsilon_{\mathcal{A}} \leq \mathsf{Adv}_{\mathcal{M}^\top\text{-}\mathsf{MDDH}}(\mathcal{B}) + 1/p$,*

where $\mathcal{M}^\top$ is the distribution which results from sampling matrices from $\mathcal{M}$ and transposing them.

Proof. (Lemma 3.1.) The proof is a direct consequence of the fact that scheme from Fig. 2 is an argument of knowledge in the generic group model, as proven by Fauzi et al. [5, Theorem 2]. Indeed, if this is the case there exists an extractor which given $\mathcal{A}$ outputs a witness $\boldsymbol{w}^*$ such that $\left(\begin{smallmatrix} 0 \\ \boldsymbol{v} \end{smallmatrix}\right) = \left(\begin{smallmatrix} \mathbf{M} \\ \mathbf{N} \end{smallmatrix}\right) \boldsymbol{w}^*$. Since $\boldsymbol{v} \neq 0$, then $\boldsymbol{w}^* \neq 0$ and $\boldsymbol{w}^* \in \mathbb{Z}_p^n$ is a non-trivial element in the kernel of $\mathbf{M}$, breaking the $\mathcal{M}$-$\mathsf{FindRep}$ assumption[6].

Proof. (Lemma 3.2). The proof follows from the indistinguishability of the following games

Game_0**:** This game runs the adversary as in Lemma 3.

Game_1**:** This game is exactly as Game_0 but the crs is computed using algorithm K^*, as defined in Fig. 3, and the winning condition is

$$\boldsymbol{v} \neq 0 \text{ and } \boldsymbol{\pi} = (\boldsymbol{v}^\top (\mathbf{C}_2 - \mathbf{K}_{2,2} \underline{\mathbf{A}}) \overline{\mathbf{A}}^{-1}, \boldsymbol{v}^\top \mathbf{K}_{2,2}),$$

where $\underline{\mathbf{A}}$ is the last row of $\mathbf{A}$ and $\overline{\mathbf{A}}$ is the first $k \times k$ block of $\mathbf{A}$.

Game_2**:** This game is exactly as Game_1 but $\boldsymbol{z} \leftarrow \mathbb{Z}_p^n$.

We now prove some Lemmas which show that the games are indistinguishable. Lemmas 4 and 5 show that the adversary has essentially the same advantage

[6] For the distribution $\mathcal{M}^\top$ used in Sect. 5 this assumption is equivalent to the m-DLog assumption.

```
$\underline{\mathsf{K}^*(gk, [\mathbf{M}]_1, [\mathbf{N}]_1):}$ // $\mathbf{M} \in \mathbb{Z}_p^{\ell_1 \times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}$
    Sample $\mathbf{A} \leftarrow \mathcal{D}_k$;
    $\mathbf{C}_1 \leftarrow \mathbb{Z}_p^{\ell_1 \times k}$; $\mathbf{C}_2 \leftarrow \mathbb{Z}_p^{\ell_2 \times k}$; $\mathbf{C} = \begin{pmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{pmatrix}$; $\mathbf{K}_{1,2} \leftarrow \mathbb{Z}_p^{\ell_1}$; $\mathbf{K}_{2,2} \leftarrow \mathbb{Z}_p^{\ell_2}$;
    $\mathbf{K}_{2,1} = (\mathbf{C}_2 - \mathbf{K}_{2,2}\underline{\mathbf{A}})\overline{\mathbf{A}}^{-1} \in \mathbb{Z}_p^{\ell_2 \times k}$; $[\boldsymbol{z}]_1 = [\mathbf{M}^\top]_1 \mathbf{K}_{1,2}$;
    $[\mathbf{B}]_1 = ([\mathbf{M}^\top \mathbf{C}_1 \overline{\mathbf{A}}^{-1} - \boldsymbol{z}\underline{\mathbf{A}}\overline{\mathbf{A}}^{-1} + \mathbf{N}^\top \mathbf{K}_{2,1}]_1, [\boldsymbol{z}]_1 + [\mathbf{N}^\top]_1 \mathbf{K}_{2,2})$;
    return crs $= (gk, [\mathbf{B}]_1, [\mathbf{A}]_2, [\mathbf{C}]_2)$.
```

Fig. 3. The modified crs generation algorithm used in Lemma 3.

of winning in any game. Lemma 6 says that the adversary has negligible probability of winning in Game_2. Lemma 3.2 follows from the composition of lemmas 4, 5 and 6.

Lemma 4. *For any (unbounded) algorithm $\mathcal{A}$ we have* $\Pr[\mathsf{Game}_1(\mathcal{A}) = 1] = \Pr[\mathsf{Game}_0(\mathcal{A}) = 1]$.

Proof. If we define $\mathbf{K}_{1,1} = (\mathbf{C}_1 - \mathbf{K}_{1,2}\underline{\mathbf{A}})\overline{\mathbf{A}}^{-1}$ and $\mathbf{K} = \begin{pmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{K}_{1,1} & \mathbf{K}_{1,2} \\ \mathbf{K}_{2,1} & \mathbf{K}_{2,2} \end{pmatrix}$, we observe that the output of K^* is well formed and the winning condition is the same as in the previous game, since

$$[\mathbf{B}]_1 = ([\mathbf{M}^\top \mathbf{C}_1 \overline{\mathbf{A}}^{-1} - \boldsymbol{z}\underline{\mathbf{A}}\overline{\mathbf{A}}^{-1} + \mathbf{N}^\top \mathbf{K}_{2,1}]_1, [\boldsymbol{z}]_1 + [\mathbf{N}^\top]_1 \mathbf{K}_{2,2})$$
$$= ([\mathbf{M}^\top \mathbf{K}_{1,1} + \mathbf{N}^\top \mathbf{K}_{2,1}]_1, [\mathbf{M}^\top \mathbf{K}_{1,2} + \mathbf{N}^\top \mathbf{K}_{2,2}]_1) = [\mathbf{M}^\top \mathbf{K}_1 + \mathbf{N}^\top \mathbf{K}_2]_1, \qquad \text{and}$$

$$\mathbf{K}\mathbf{A} = \begin{pmatrix} (\mathbf{C}_1 - \mathbf{K}_{1,2}\underline{\mathbf{A}})\overline{\mathbf{A}}^{-1} & \mathbf{K}_{1,2} \\ (\mathbf{C}_2 - \mathbf{K}_{2,2}\underline{\mathbf{A}})\overline{\mathbf{A}}^{-1} & \mathbf{K}_{2,2} \end{pmatrix} \begin{pmatrix} \overline{\mathbf{A}} \\ \underline{\mathbf{A}} \end{pmatrix} = \begin{pmatrix} \mathbf{C}_1 - \mathbf{K}_{1,2}\underline{\mathbf{A}} + \mathbf{K}_{1,2}\underline{\mathbf{A}} \\ \mathbf{C}_2 - \mathbf{K}_{2,2}\underline{\mathbf{A}} + \mathbf{K}_{2,2}\underline{\mathbf{A}} \end{pmatrix} = \mathbf{C},$$

and by definition $\boldsymbol{\pi} = (\boldsymbol{v}^\top (\mathbf{C}_2 - \mathbf{K}_{2,2}\underline{\mathbf{A}})\overline{\mathbf{A}}^{-1}, \boldsymbol{v}^\top \mathbf{K}_{2,2}) = (\boldsymbol{v}^\top \mathbf{K}_{2,1}, \boldsymbol{v}^\top \mathbf{K}_{2,2}) = \boldsymbol{v}^\top \mathbf{K}_2$.

Therefore we just need to argue that the distribution of $\mathbf{K}$ is the same in both games. But this is an immediate consequence of the fact that for every value of $(\mathbf{C}, \mathbf{K}_{1,1}, \mathbf{K}_{2,1})$ there exists a unique value of $(\mathbf{K}_{1,2}, \mathbf{K}_{2,2})$ which is compatible with $\mathbf{C} = \mathbf{K}\mathbf{A}$. Indeed, $\mathbf{C} = \mathbf{K}\mathbf{A} \iff \mathbf{C}_i = \mathbf{K}_{i,1}\overline{\mathbf{A}} + \mathbf{K}_{i,2}\underline{\mathbf{A}},\ i = 1, 2 \iff (\mathbf{C}_i - \mathbf{K}_{i,2}\underline{\mathbf{A}})\overline{\mathbf{A}}^{-1} = \mathbf{K}_{i,1},\ i = 1, 2$.

Lemma 5. *For any PPT algorithm $\mathcal{A}$ there exists a PPT algorithm $\mathcal{B}$ such that* $|\Pr[\mathsf{Game}_1(\mathcal{A}) = 1] - \Pr[\mathsf{Game}_0(\mathcal{A}) = 1]| \leq \mathsf{Adv}_{\mathcal{M}^\top\text{-}\mathsf{MDDH}}(\mathcal{B})$.

Proof. We construct an adversary $\mathcal{B}$ that receives the challenge $([\mathbf{M}^\top]_1, [\boldsymbol{z}^*]_1)$, where $\boldsymbol{z}^*$ is either $\mathbf{M}^\top \boldsymbol{r}$, $\boldsymbol{r} \leftarrow \mathbb{Z}_p^{\ell_1}$, or $\boldsymbol{z}^* \leftarrow \mathbb{Z}_p^n$. $\mathcal{B}$ computes the crs running $\mathsf{K}^*(gk, [\mathbf{M}]_1, [\mathbf{N}]_1)$ but replaces $[\boldsymbol{z}]_1$ with $[\boldsymbol{z}^*]_1$, and then runs $\mathcal{A}$ as in game Game_1. It follows that $\Pr[\mathcal{B}([\mathbf{M}^\top]_1, [\boldsymbol{z}^*]_1) = 1 | \boldsymbol{z}^* = \mathbf{M}^\top \boldsymbol{r}] = \Pr[\mathsf{Game}_1(\mathcal{A}) = 1]$ and $\Pr[\mathcal{B}([\mathbf{M}^*]_1, [\boldsymbol{z}^*]_1) = 1 | \boldsymbol{z}^* \leftarrow \mathbb{Z}_p^n] = \Pr[\mathsf{Game}_2(\mathcal{A}) = 1]$ and the lemma follows.

Lemma 6. *For any (unbounded) algorithm $\mathcal{A}$,* $\Pr[\mathsf{Game}_2(\mathcal{A}) = 1] \leq 1/p$.

Proof. We will show that, conditioned on $\mathbf{A}, \mathbf{C}, \mathbf{B}, \mathbf{M}, \mathbf{N}$, the matrix $\mathbf{K}_{2,2}$ is uniformly distributed. Since it holds that $\mathbf{BA} = (\mathbf{M}^\top, \mathbf{N}^\top)\mathbf{C}$, we get that the first k columns of $\mathbf{B}$, namely $\mathbf{B}_1$, are completely determined by $\mathbf{B}_2$, the last column of $\mathbf{B}$. Indeed

$$(\mathbf{B}_1, \mathbf{B}_2)\mathbf{A} = (\mathbf{M}^\top, \mathbf{N}^\top)\mathbf{C} \iff \mathbf{B}_1 = ((\mathbf{M}^\top, \mathbf{N}^\top)\mathbf{C} - \mathbf{B}_2\underline{\mathbf{A}})\overline{\mathbf{A}}^{-1}.$$

Hence, conditioning in $\mathbf{A}, \mathbf{C}, \mathbf{B}_2, \mathbf{M}, \mathbf{N}$ doesn't alter the probability. We have that $\mathbf{B}_2 = \boldsymbol{z} + \mathbf{N}^\top \mathbf{K}_{2,2}$, which consists of n equations on $n + \ell_2$ variables. It follows that there are ℓ_2 free variables. Then $\mathbf{K}_{2,2}$ is uniformly distributed and hence completely hidden to the adversary.

Note that

$$\boldsymbol{\pi} = \boldsymbol{v}^\top \mathbf{K}_2 \Longrightarrow \boldsymbol{\pi}_2 = \boldsymbol{v}^\top \mathbf{K}_{2,2},$$

where $\boldsymbol{\pi}_2$ is the last element of $\boldsymbol{\pi}$. Given that $\boldsymbol{v} \neq 0$, the last equation only holds with probability $1/p$ and so $\mathcal{A}$'s probability of winning.

The knowledge transfer property is a direct consequence of Lemma 3.

Theorem 1. *For any adversary $\mathcal{A}$ against the soundness of* Lin *with respect to $\mathcal{L}_{NO}^{\mathsf{lin}}$, it holds that:*

1. *When $\widetilde{\mathcal{D}}_k = \overline{\mathcal{D}}_k$, $\mathcal{M}$ is witness samplable, if $\mathcal{A}$ is generic then there exists a PPT adversary $\mathcal{B}$ such that $\epsilon_{\mathcal{A}} \leq \mathsf{Adv}_{\mathcal{M}\text{-}\mathsf{FindRep}}(\mathcal{B}) + \mathsf{negl}(\lambda)$.*
2. *When $\widetilde{\mathcal{D}}_k = \mathcal{D}_k$, there exist adversaries $\mathcal{B}_1$ and $\mathcal{B}_2$ such that*

$$\mathsf{Adv}_{\mathsf{Lin}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathcal{D}_k\text{-}\mathsf{KerMDH}}(\mathcal{B}_1) + \mathsf{Adv}_{\mathcal{M}^\top\text{-}\mathsf{MDDH}}(\mathcal{B}_2) + 1/p.$$

Proof. Both for the witness samplable and the general case, given an adversary that produces a valid proof for a statement in $\mathcal{L}_{NO}^{\mathsf{lin}}$, successful attacks can be divided in two categories.

Type I: In this attack $[\boldsymbol{\pi}]_1 \neq [\boldsymbol{u}^\top]_1\mathbf{K}_1 + [\boldsymbol{v}^\top]_1\mathbf{K}_2$.
Type II: In this type of attack $[\boldsymbol{\pi}]_1 = [\boldsymbol{u}^\top]_1\mathbf{K}_1 + [\boldsymbol{v}^\top]_1\mathbf{K}_2$.

Type I attacks are not possible when $\overline{k} = k$, because proofs are unique, i.e. there is only one value of $\boldsymbol{\pi}$ which can satisfy the verification equation. Type I attacks are computationally infeasible when $\overline{k} = k + 1$, as they can be used to construct an adversary $\mathcal{B}_1$ against the $\mathcal{D}_k$-KerMDH assumption.[7] Adversary $\mathcal{B}_1$ receives a challenge $[\mathbf{A}]_2$ and then runs the soundness experiment for $\mathcal{A}$. When $\mathcal{A}$ outputs $([\boldsymbol{u}]_1, [\boldsymbol{v}]_1, [\boldsymbol{\pi}]_1)$, $\mathcal{B}_1$ outputs $[\boldsymbol{\pi}^\dagger]_1 = [\boldsymbol{\pi}]_1 - [\boldsymbol{u}^\top]_1\mathbf{K}_1 - [\boldsymbol{v}^\top]_1\mathbf{K}_2 \neq 0$. Since $[\boldsymbol{\pi}]_1$ is accepted by the verifier we get that $e([\boldsymbol{\pi}]_1, [\mathbf{A}]_2) = e([\boldsymbol{u}^\top]_1, [\mathbf{C}_1]_2) + e([\boldsymbol{v}^\top]_1, [\mathbf{C}_2]_2)$ and then $\boldsymbol{\pi}^\dagger \mathbf{A} = \boldsymbol{\pi}\mathbf{A} - \boldsymbol{u}^\top \mathbf{K}_1 \mathbf{A} - \boldsymbol{v}^\top \mathbf{K}_2 \mathbf{A} = \boldsymbol{\pi}\mathbf{A} - \boldsymbol{u}^\top \mathbf{C}_1 -$

[7] This part of the proof follows essentially the same lines of the first constant-size QA-NIZK arguments for linear spaces of Libert et al.[26] which were later simplified and generalized by Kiltz and Wee [25].

$\boldsymbol{v}^\top\mathbf{C}_2 = 0$. We conclude that the success probability of a type I attack is bounded by $\mathsf{Adv}_{\mathcal{D}_k\text{-}\mathsf{KerMDH}}(\mathcal{B}_1)$.

For type II attacks, for both types of distributions, since $[\boldsymbol{\pi}]_1 = [\boldsymbol{u}^\top]_1\mathbf{K}_1 + [\boldsymbol{v}^\top]_1\mathbf{K}_2$ is a valid proof for $\begin{pmatrix}[\boldsymbol{u}]_1\\ [\boldsymbol{v}]_1\end{pmatrix}$, then, by linearity of the verification equation, $\boldsymbol{\pi}^\dagger = \boldsymbol{\pi} - \boldsymbol{w}^\top\mathbf{B}$ is a valid proof for $\begin{pmatrix}0\\ [\boldsymbol{v}^\dagger]_1\end{pmatrix} = \begin{pmatrix}[\boldsymbol{u}]_1-[\mathbf{M}]_1\boldsymbol{w}\\ [\boldsymbol{v}]_1-[\mathbf{N}]_1\boldsymbol{w}\end{pmatrix}$. Since $\boldsymbol{v} \neq \mathbf{N}\boldsymbol{w}$, we conclude that an attacker of type II can be turned into an attacker $\mathcal{B}_2$ for Lemma 3.

4.3 Extension to SMDDH Assumptions

In Sect. 5 the crs includes $\mathbf{M}$ in both groups, i.e. $[\mathbf{M}]_1, [\mathbf{M}]_2$. This implies that we need to prove Lemma 3 even when the adversary is given $[\mathbf{M}]_1, [\mathbf{M}]_2$. But this is not a problem, since we can build an adversary for Lemma 5 against the $\mathcal{M}^\top$-$\mathsf{SMDDH}_{\mathbb{G}_1}$ assumption. Similarly, we can prove that Theorem 1 holds, even when the adversary is given $[\mathbf{M}]_1, [\mathbf{M}]_2$, assuming the hardness of the $\mathcal{M}^\top$-SMDDH assumption.

4.4 Extension to Bilateral Linear Spaces

In Sect. 5 we need a QA-NIZK argument for bilateral linear spaces [14], which are linear spaces split between $\mathbb{G}_1$ and $\mathbb{G}_2$. In [14], a QA-NIZK argument for such languages is given, which is very close to the argument of membership in (unilateral) linear spaces of [25]. In Fig. (4) we describe the QA-NIZK argument of [14] adapted to matrices with 3 blocks. The proof of the knowledge transfer property is essentially the same as in the unilateral case and can be found in the full version of this work.

5 A New Argument for Correct Arithmetic Circuit Evaluation

In this section we describe our construction for proving correct evaluation of an arithmetic circuit. It makes use of two subarguments: a quadratic and a linear "knowledge transfer" subarguments. The reason why we use the term "knowledge transfer" is because these arguments will ensure that, if the prover knows a witness for the circuit evaluation up to level i which is also a valid opening up to level i of a set of shrinking commitments to the corresponding wires, it also knows a valid opening to the commitments of the wires at level $i+1$.

Since the input of the circuit is public, the idea is that these arguments allow to "transfer" the knowledge of the witness for correct evaluation (a consistent assignment to all wires) to lower levels of the circuit. Any adversary against soundness needs to break the "chain" of consistent evaluations at some point and thus, break the soundness of one of the two subarguments. This technique allows us to avoid using binding commitments to the wires at each level, while

```
$\underline{\mathsf{K}(gk, [\mathbf{M}]_1, [\mathbf{N}]_1, [\mathbf{P}]_2)}$:
// $\mathbf{M} \in \mathbb{Z}_p^{\ell_1 \times n}, \mathbf{N} \in \mathbb{Z}_p^{\ell_2 \times n}, \mathbf{P} \in \mathbb{Z}_p^{\ell_3 \times n}$
    $\mathbf{K}_1 \leftarrow \mathbb{Z}_p^{\ell_1 \times \bar{k}}$; $\mathbf{K}_2 \leftarrow \mathbb{Z}_p^{\ell_2 \times \bar{k}}$; $\mathbf{K}_3 \leftarrow \mathbb{Z}_p^{\ell_3 \times \bar{k}}$
    $\mathbf{K}^\top = \left(\mathbf{K}_1^\top, \mathbf{K}_2^\top, \mathbf{K}_3^\top\right)$;
    Sample $\mathbf{A} \leftarrow \widetilde{\mathcal{D}}_k$; $\mathbf{\Gamma} \leftarrow \mathbb{Z}_p^{n \times \bar{k}}$
    $[\mathbf{B}]_1 = [\mathbf{M}^\top \mathbf{K}_1 + \mathbf{N}^\top \mathbf{K}_2 + \mathbf{\Gamma}]_1$;
    $[\mathbf{D}]_2 = [\mathbf{P}^\top \mathbf{K}_3 - \mathbf{\Gamma}]$;
    $\mathbf{C}_1 = \mathbf{K}_1 \mathbf{A}$; $\mathbf{C}_2 = \mathbf{K}_2 \mathbf{A}$;
    $\mathbf{C}_3 = \mathbf{K}_3 \mathbf{A}$: $\mathbf{C} = \mathbf{K} \mathbf{A}$
    return $\mathrm{crs} = (gk, [\mathbf{B}]_1, [\mathbf{D}]_2, [\mathbf{A}]_{1,2},$
    $[\mathbf{C}_1]_2, [\mathbf{C}_2]_2, [\mathbf{C}_3]_1)$.
```

```
$\underline{\mathsf{P}(\mathrm{crs}, [\boldsymbol{u}]_1, [\boldsymbol{v}_1]_1, [\boldsymbol{v}_2]_2, \boldsymbol{w})}$:
    $\boldsymbol{\rho} \leftarrow \mathbb{Z}_p^{\bar{k}}$;
    $[\boldsymbol{\pi}]_1 = \boldsymbol{w}^\top [\mathbf{B}]_1 + [\boldsymbol{\rho}]_1$;
    $[\boldsymbol{\theta}]_1 = \boldsymbol{w}^\top [\mathbf{D}]_2 - [\boldsymbol{\rho}]_2$;
    return $([\boldsymbol{\pi}]_1, [\boldsymbol{\theta}]_2)$.
$\underline{\mathsf{V}(\mathrm{crs}, [\boldsymbol{u}]_1, [\boldsymbol{v}_1]_1, [\boldsymbol{v}_2]_2, [\boldsymbol{\pi}]_1, [\boldsymbol{\theta}]_2)}$:
    Check if:
    $e([\boldsymbol{\pi}]_1, [\mathbf{A}]_2) - e([\boldsymbol{u}^\top]_1, [\mathbf{C}_1]_2)$
    $-e([\boldsymbol{v}_1^\top]_1, [\mathbf{C}_2]_2) =$
    $e([\boldsymbol{\theta}]_2, [\mathbf{A}]_1) - e([\boldsymbol{v}_2^\top]_2, [\mathbf{C}_3]_1)$
```

Fig. 4. The $\mathsf{BLin}_{\widetilde{\mathcal{D}}_k}$ argument for proving membership in bilateral linear spaces. The matrix $\mathbf{A}$ is either sampled from a distribution $\widetilde{\mathcal{D}}_k = \overline{\mathcal{D}}_k$ or from a distribution $\widetilde{\mathcal{D}}_k = \mathcal{D}_k$, such that the $\mathcal{D}_k$-$\mathsf{SKerMDH}$ assumption holds. In the latter case $\bar{k} = k + 1$ while in former case $\bar{k} = k$. Since the $\mathcal{D}_1$-$\mathsf{SKerMDH}$ is false [14] for any $\mathcal{D}_1$, it should hold that $k \geq 2$.

still being able to define what it means to break soundness. Intuitively, the difficulty we have to circumvent is to reason about whether the openings of shrinking commitments satisfy a certain equation without assuming that the adversary is generic, as there are many possible such openings.

The reason why we use two arguments is natural given characterization of circuits given in Sect. 3. The variables A_{ij} (resp. B_{ij}, C_{ij}) describe correct assignments to the j-th left (resp. right, output) wire at level i. We use the quadratic knowledge transfer property to ensure that a certain value $\boldsymbol{O}_i$ is a valid (deterministic, not hiding) commitment to all the outputs at level i if $\boldsymbol{L}_{i-1}$ and $\boldsymbol{R}_{i-1}$ are valid commitments (i.e. consistent with the input) to all the right and left wires at the previous level. On the other hand, we encode the affine constraints as membership in linear spaces and use the linear knowledge transfer argument to ensure that $\boldsymbol{L}_i, \boldsymbol{R}_i$ are valid commitments to all left and right wires at level i if $\boldsymbol{O}_j$ for $j = 1, \ldots, i-1$ are valid commitments to the previous levels.

Throughout this section, R_ϕ represents a relation $R_\phi = \{(gk, \boldsymbol{x}, \boldsymbol{y}) : \phi(\boldsymbol{x}) = \boldsymbol{y}\}$ where gk is an asymmetric bilinear group of order p and $\phi : \mathbb{Z}_p^{n_0} \to \mathbb{Z}_p^{n_d}$ as described in Sect. 3 and $N = \max_{i=1,\ldots,d} n_i$ is the maximum number of multiplicative gates of same multiplicative depth. The construction is parameterized by a value k_s, following the dicussion in Sect. 4.2 on the security properties of the linear knowledge transfer argument.

This section is organized as follows: we first show how to encode affine constraints as membership in linear spaces, then we present the description of our argument in terms of the two subarguments and give the (sketched) proof of security, and finally we discuss its efficiency.

5.1 Encoding Affine Constraints as Membership in Linear Spaces

We translate the affine constraints described in the circuit encoding of Sect. 3 as membership of $([\boldsymbol{O}]_1, [\boldsymbol{L}]_1, [\boldsymbol{R}]_2)$ in a linear subspace of $\mathbb{G}_1^{n+(2d-1)k_s} \times \mathbb{G}_2^{dk_s}$.

We write in matrix form the expression of $(\boldsymbol{x}, [\boldsymbol{O}]_1, [\boldsymbol{L}]_1, [\boldsymbol{R}]_2)$ in terms of the internal wires of the circuit, following Sect. 3. The commitments to the output values $[\boldsymbol{O}]_1$ should satisfy that $[\boldsymbol{O}_i]_1 = [\boldsymbol{\Lambda}_i]_1 \boldsymbol{c}_i$, where $\boldsymbol{\Lambda}_i = (\lambda_1(\boldsymbol{s}), \ldots, \lambda_{n_i}(\boldsymbol{s}))$ and $\lambda_j(X)$ is the jth Lagrangian polynomial for some $\mathcal{R} = \{r_1, \ldots, r_N\} \subset \mathbb{Z}_p$ and the input $\boldsymbol{x} = \boldsymbol{c}_0$ is public. These constraints can be expressed in matrix form in equation (4):

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{O}_1 \\ \boldsymbol{O}_2 \\ \boldsymbol{O}_3 \\ \vdots \\ \boldsymbol{O}_{d-1} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_1 & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_2 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda}_3 & & \mathbf{0} \\ \vdots & \vdots & \vdots & & \ddots & \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \boldsymbol{\Lambda}_{d-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{c}_0 \\ \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \boldsymbol{c}_3 \\ \vdots \\ \boldsymbol{c}_{d-1} \end{pmatrix} \tag{4}$$

We denote the matrix on the right hand side of (4) as $\mathbf{M}$, so this equation reads $\left(\begin{smallmatrix} \boldsymbol{x} \\ \boldsymbol{O} \end{smallmatrix}\right) = \mathbf{M}\boldsymbol{c}$. On the other hand, the constraints satisfied by the left wires in terms of the output wires of previous levels can be written in matrix form as shown in equation (5):

$$\begin{pmatrix} \boldsymbol{L}_1 \\ \boldsymbol{L}_2 \\ \boldsymbol{L}_3 \\ \vdots \\ \boldsymbol{L}_d \end{pmatrix} = \begin{pmatrix} \mathbf{F}_{1,0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{F}_{2,0} & \mathbf{F}_{2,1} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{F}_{3,0} & \mathbf{F}_{3,1} & \mathbf{F}_{3,2} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}_{d,0} & \mathbf{F}_{d,1} & \mathbf{F}_{d,2} & \ldots & \mathbf{F}_{d,d-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{c}_0 \\ \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_{d-1} \end{pmatrix} + \begin{pmatrix} \hat{\boldsymbol{L}}_1 \\ \hat{\boldsymbol{L}}_2 \\ \hat{\boldsymbol{L}}_3 \\ \vdots \\ \hat{\boldsymbol{L}}_d \end{pmatrix}, \tag{5}$$

that is, for each i, $\boldsymbol{L}_i = \sum_{k=0}^{i-1} \mathbf{F}_{i,k}\boldsymbol{c}_k + \hat{\boldsymbol{L}}_i$, where

$$\begin{aligned} \mathbf{F}_{i,k} &= \left(\textstyle\sum_{j=1}^{n_k} f_{ijk1}\lambda_j(\boldsymbol{s}),\, \sum_{j=1}^{n_k} f_{ijk2}\lambda_j(\boldsymbol{s}),\, \ldots\, \sum_{j=1}^{n_k} f_{ijkn_k}\lambda_j(\boldsymbol{s})\right) \\ &= \left(v_{ik1}(\boldsymbol{s}),\, v_{ik2}(\boldsymbol{s}),\, \ldots\, v_{ikn_k}(\boldsymbol{s})\right) \end{aligned} \tag{6}$$

and $\hat{\boldsymbol{L}}_i = \sum_{j=1}^{n_i} f_{ij}\lambda_j(\boldsymbol{s}) = v_i(\boldsymbol{s})$, for the constants which are defined in Lemma 1. We denote the matrix on the right hand side of equation (5) as $\mathbf{N}$, so this equation reads $\boldsymbol{L} = \mathbf{N}\boldsymbol{c} + \hat{\boldsymbol{L}}$. The constraints satisfied by the right wires in terms of the output wires of previous levels can be written in a similar form as shown in equation (7):

$$\begin{pmatrix} \boldsymbol{R}_1 \\ \boldsymbol{R}_2 \\ \boldsymbol{R}_3 \\ \vdots \\ \boldsymbol{R}_d \end{pmatrix} = \begin{pmatrix} \mathbf{G}_{1,0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{G}_{2,0} & \mathbf{G}_{2,1} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{G}_{3,0} & \mathbf{G}_{3,1} & \mathbf{G}_{3,2} & \ldots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_{d,0} & \mathbf{G}_{d,1} & \mathbf{G}_{d,2} & \ldots & \mathbf{G}_{d,d-1} \end{pmatrix} \begin{pmatrix} \boldsymbol{c}_0 \\ \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_{d-1} \end{pmatrix} + \begin{pmatrix} \hat{\boldsymbol{R}}_1 \\ \hat{\boldsymbol{R}}_2 \\ \hat{\boldsymbol{R}}_3 \\ \vdots \\ \hat{\boldsymbol{R}}_d \end{pmatrix}, \tag{7}$$

that is, for each i, $\boldsymbol{R}_i = \sum_{k=0}^{i-1} \mathbf{G}_{i,k}\boldsymbol{c}_k + \hat{\boldsymbol{R}}_i$, where

$$\begin{aligned}\mathbf{G}_{i,k} &= \left(\sum_{j=1}^{n_k} g_{ijk1}\lambda_j(\boldsymbol{s}), \sum_{j=1}^{n_k} g_{ijk2}\lambda_j(\boldsymbol{s}), \dots \sum_{j=1}^{n_k} g_{ijkn_k}\lambda_j(\boldsymbol{s})\right) \\ &= \left(w_{ik1}(\boldsymbol{s}), w_{ik2}(\boldsymbol{s}), \dots w_{ikn_k}(\boldsymbol{s})\right), \qquad (8)\end{aligned}$$

and $\hat{\boldsymbol{R}}_i = \sum_{j=1}^{n_i} g_{ij}\lambda_j(\boldsymbol{s}) = w_i(\boldsymbol{s})$. We denote the matrix on the right hand side of equation (7) as $\mathbf{P}$, so this equation reads $\boldsymbol{R} = \mathbf{P}\boldsymbol{z} + \hat{\boldsymbol{R}}$.

With the notation defined, satisfaction of the affine constraints can be written as $\begin{pmatrix} [\boldsymbol{O}']_1 \\ [\boldsymbol{L}]_1 - [\hat{\boldsymbol{L}}]_1 \\ [\boldsymbol{R}]_2 - [\hat{\boldsymbol{R}}]_2 \end{pmatrix} \in \mathbf{Im}\begin{pmatrix} [\mathbf{M}]_1 \\ [\mathbf{N}]_1 \\ [\mathbf{P}]_2 \end{pmatrix}$, where $[\boldsymbol{O}']_1 = \begin{pmatrix} [\boldsymbol{x}]_1 \\ [\boldsymbol{O}]_1 \end{pmatrix}$. That is, the linear constraints are satisfied if a certain vector is in a subspace generated by some matrix which depends on the circuit.

5.2 New Argument

In this section we describe our construction for proving correct evaluation of an arithmetic circuit.

$\mathsf{Setup}(R_\phi)$**:** Pick $\boldsymbol{s} \leftarrow \mathbb{Z}_p^{k_s}$. Generate $\mathrm{crs}_\phi = (crs_{\phi,1}, \dots, \mathrm{crs}_{\phi,k_s})$, where $\mathrm{crs}_{\phi,i} \leftarrow \mathsf{Quad.K}(gk, \{[s_i^j]_1\}_{j=1}^{N-1}, \{[s_i^j]_2\}_{j=1}^{N})$ is the crs for the quadratic knowledge transfer argument defined in Fig. 1. Express affine constraints (equations 4),(5), and (7)) which define circuit satisfiability as membership in the image of $([\mathbf{M}^\top]_1, [\mathbf{N}^\top]_1, [\mathbf{P}^\top]_2)^\top$ as explained in Sect. 5.1. Generate a crs for the bilateral linear knowledge transfer argument defined in Fig. 4 for $([\mathbf{M}^\top]_1, [\mathbf{N}^\top]_1, [\mathbf{P}^\top]_2)^\top$.

$\mathsf{Prove}(\mathrm{crs}, (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}) \in \mathcal{R}_\phi)$**:** Given the input $\boldsymbol{x}$, the output $\boldsymbol{y}$, and $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ a valid assignment to left, right and output wires as described in Lemma 1, the prover proceeds as follows:

1. For each $i \in \{1, \dots, d\}$, commit to $\boldsymbol{a}_i, \boldsymbol{c}_i$ in $\mathbb{G}_1^{k_s}$ and to $\boldsymbol{b}_i$ in $\mathbb{G}_2^{k_s}$ as: $[\boldsymbol{L}_i]_1 = \sum_{j=1}^{n_i} a_{ij}[\lambda_j(\boldsymbol{s})]_1 = [\boldsymbol{\Lambda}_i]_1\boldsymbol{a}_i$, $[\boldsymbol{R}_i]_2 = \sum_{j=1}^{n_i} b_{i,j}[\lambda_j(\boldsymbol{s})]_2 = [\boldsymbol{\Lambda}_i]_2\boldsymbol{b}_i$, $[\boldsymbol{O}_i]_1 = \sum_{j=1}^{n_i} c_{ij}[\lambda_j(\boldsymbol{s})]_1 = [\boldsymbol{\Lambda}_i]_1\boldsymbol{c}_i$.
2. (Quadratic Constraints) For each $i \in \{1, \dots, d\}$, and each $j \in \{1, \dots, k_s\}$, compute a proof $\mathit{\Pi}_{i,j}^{\mathsf{quad}}$ that the vector $\boldsymbol{a}_i \circ \boldsymbol{b}_i$, which is the componentwise product of the openings of $[L_{ij}]_1, [R_{ij}]_2$, is an opening of $[O_{ij}]_1$.
3. (Linear Constraints) Compute a proof $\mathit{\Pi}^{\mathsf{lin}}$ that $[\boldsymbol{L}_i]_1$ and $[\boldsymbol{R}_i]_2$ are commitments to the correct evaluation of all the left and right wires at level i, for all $i \in \{1, \dots, d\}$, that is, that they satisfy the affine linear constraints which relate them to the outputs of gates at levels $j = 0, \dots, i-1$.
4. Output $(\mathcal{C} = ([\boldsymbol{L}]_1, [\boldsymbol{R}]_2, [\boldsymbol{O}]_1), \mathit{\Pi}^{\mathsf{quad}}, \mathit{\Pi}^{\mathsf{lin}})$ as the proof, where $\mathit{\Pi}^{\mathsf{quad}} = \{\mathit{\Pi}_{i,j}^{\mathsf{quad}} : i = 1, \dots, d,\ j = 1, \dots, k_s\}$.

$\mathsf{Verify}(\mathrm{crs}, (\boldsymbol{x}, \boldsymbol{y}), (\mathcal{C}, \mathit{\Pi}^{\mathsf{quad}}, \mathit{\Pi}^{\mathsf{lin}}))$**:** Output 1 if the following two checks are successful and 0 otherwise:

1. Verify $\mathit{\Pi}^{\mathsf{quad}}, \mathit{\Pi}^{\mathsf{lin}}$.
2. Check that $[\boldsymbol{O}_d]_1 = \sum_{j=1}^{n_d} [\lambda_j(\boldsymbol{s})]_1 y_j$.

Security. Perfect completeness is obvious, because if $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ is a valid witness for satisfiability, then it satisfies both linear and quadratic constraints because of the characterization of Sect. 3 and the definition of $\mathbf{M}, \mathbf{N}, \mathbf{P}$ presented in Sect. 5.1.

Let $\mathcal{A}$ be an adversary against the soundness of the scheme. We construct an adversary $\mathcal{B}_1$ against the quadratic knowledge transfer argument, $\mathcal{B}_{2,0}, \ldots, \mathcal{B}_{2,d-1}$ against the linear knowledge transfer argument.

Adversary $\mathcal{B}_1$ receives the common reference string of the quadratic subargument, which includes $(gk, \left\{[s^i]_1\right\}_{i=1}^{N-1}, \left\{[s^i]_2\right\}_{i=1}^{N})$ and samples $\alpha_j \leftarrow \mathbb{Z}_p^*$, $j = 2, \ldots, k_s$. It defines $s = s_1$, $s_j = \alpha_j s_j$ and computes the crs of the quadratic argument for s_j, $j = 1, \ldots, k_s$ from the received values. It then creates the common reference string of the full argument in the natural way, by defining the matrices $\mathbf{M}, \mathbf{N}, \mathbf{P}$ from the crs of the quadratic subargument and sampling the rest of the secret key. When it receives an accepting proof $(\mathcal{C} = ([\boldsymbol{L}]_1, [\boldsymbol{R}]_2, [\boldsymbol{O}]_1), \mathit{\Pi}^{\mathsf{quad}}, \mathit{\Pi}^{\mathsf{lin}})$ from adversary $\mathcal{A}$ for some statement $(\boldsymbol{x}, \boldsymbol{y})$, adversary $\mathcal{B}_1$ computes the full witness for correct evaluation $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ from $\boldsymbol{x}$. The adversary searches for indexes i, j such that $[L_{ij}]_1$ and $[R_{ij}]_2$ are commitments to $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ but $[O_{ij}]_1$ is not a valid commitment to $\boldsymbol{a}_i \circ \boldsymbol{b}_i$, and it aborts if these indexes do not exist. From α_j, adversary $\mathcal{A}$ computes $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{n_i}) \in \mathbb{Z}_p^{n_i}$ such that $\lambda_\ell(s_j) = \mu_\ell \lambda_\ell(s)$ and $\nu \in \mathbb{Z}_p$ such that $\nu t(s_j) = t(s)$. It returns $(\boldsymbol{a}_i \circ \boldsymbol{\mu}, \boldsymbol{b}_i \circ \boldsymbol{\mu}, [L_{ij}]_1, [R_{ij}]_2, [O_{ij}]_1)$, as an instance of $\mathcal{L}_{NO}^{\mathsf{quad}}$ together with an accepting proof $[\nu H_{ij}]_1$.

Adversary $\mathcal{B}_{2,i}$, $i = 0, \ldots, d-1$ receives a common reference string of the linear subargument for the language associated to the first $i+1$ (resp. $i+2, i+2$) blocks of rows and the first $\sum_{j=0}^{i} n_i$ columns of $\mathbf{M}$ (resp. $\mathbf{N}, \mathbf{P}$). That is, $\mathbf{M}_i, \mathbf{N}_i$ are defined as:

$$\mathbf{M}_i = \begin{pmatrix} \mathbf{I} & & & \mathbf{0} \\ & \mathbf{\Lambda}_1 & & \\ & & \ddots & \\ \mathbf{0} & & & \mathbf{\Lambda}_i \end{pmatrix}, \qquad \mathbf{N}_i = \begin{pmatrix} \mathbf{F}_{1,0} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{F}_{2,0} & \mathbf{F}_{2,1} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}_{i+1,0} & \mathbf{F}_{i+1,1} & \ldots & \mathbf{F}_{i+1,i} \end{pmatrix},$$

and $\mathbf{P}_i$ is defined similarly. Using the linear properties of the crs, $\mathcal{B}_{2,i}$ computes the common reference string of the full argument.[8] When it receives an accepting proof $(\mathcal{C} = ([\boldsymbol{L}]_1, [\boldsymbol{R}]_2, [\boldsymbol{O}]_1)\}_{i=1}^{d}, \mathit{\Pi}^{\mathsf{quad}}, \mathit{\Pi}^{\mathsf{lin}})$ from adversary $\mathcal{A}$ for some statement $(\boldsymbol{x}, \boldsymbol{y})$, adversary $\mathcal{B}_{2,i}$ computes the full witness $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$. It then checks if $[\boldsymbol{O}_1]_1, \ldots, [\boldsymbol{O}_i]_1$ are commitments to $\boldsymbol{c}_1, \ldots, \boldsymbol{c}_i$ but either $[\boldsymbol{L}_{i+1}]_1$ or $[\boldsymbol{R}_{i+1}]_2$ are not valid commitments to $\boldsymbol{a}_i$ or $\boldsymbol{b}_i$. If this is not the case, it aborts. Else it outputs $(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_i, [\boldsymbol{O}_1]_1, \ldots, [\boldsymbol{O}_i], [\boldsymbol{L}_1]_1 - [\hat{\boldsymbol{L}}_1], [\boldsymbol{L}_{i+1}]_1 - [\hat{\boldsymbol{L}}_{i+1}], [\boldsymbol{R}_1]_2 - [\hat{\boldsymbol{R}}_1]_2, \ldots, [\boldsymbol{R}_{i+1}]_2 - [\hat{\boldsymbol{R}}_{i+1}]_2)$ together with its corresponding proof, which adversary $\mathcal{B}_{2,i}$ can compute from the proof given by adversary $\mathcal{A}$ and the secret values it sampled to extend the crs of the subargument to the full crs (this is possible using the linearity of the proof, full details are in the full version of this work.

[8] We can assume w.l.o.g. that the crs for the linear knowledge transfer associated to $\mathbf{M}_i, \mathbf{N}_i, \mathbf{P}_i$ includes $\{[s_j]_{1,2}\}_{j=1}^{N-1}, [s_j^N]$, as this does not compromise security.

For every successful adversary $\mathcal{A}$ at least one of the adversaries $\mathcal{B}_1, \mathcal{B}_{2,0}, \ldots, \mathcal{B}_{2,d-1}$ does not abort. This is because if the statement is false there must be some point in the "chain" where either $[\boldsymbol{L}_i]_1, [\boldsymbol{R}_i]_2$ are honestly computed but $[\boldsymbol{O}_i]_1$ is not, or $[\boldsymbol{O}_i]_1$ is honestly computed but $[\boldsymbol{L}_{i+1}]$ or $[\boldsymbol{R}_{i+1}]$ is not.

The linear knowledge transfer argument at level i is based on the $\mathcal{L}_2$-$\mathsf{SKerMDH}$ and the $\mathcal{M}_i^\top$-$\mathsf{SMDDH}_{\mathbb{G}_1}$ assumptions. The latter reduces to the $\mathcal{LR}_{\mathcal{R},k_s}$-$\mathsf{SMDDH}_{\mathbb{G}_1}$ and the SXDH assumptions as proven in the full version of this work. Based on this proof, we can state the following Theorem.

Theorem 2. *Let $(gk, \phi : \mathbb{Z}_p^{n_0} \to \mathbb{Z}_p^{n_d}, \mathcal{R})$ be a bilinear group of order p, an arithmetic circuit and a set of $\mathbb{Z}_p$ of cardinal $N = \max_{i=1,\ldots,d} n_i$. For any adversary $\mathcal{A}$ against the soundness of the argument defined above there exist adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{B}_4$ such that:*

$$\begin{aligned}\mathsf{Adv}_{\mathsf{snd}}(\mathcal{A}) \leq & \mathsf{Adv}_{\mathcal{R}\text{-}\mathsf{RSDH}}(\mathcal{B}_1) + d\mathsf{Adv}_{\mathcal{L}_2\text{-}\mathsf{SKerMDH}}(\mathcal{B}_2) + dk_s\mathsf{Adv}_{\mathcal{LG}_{\mathcal{R},k_s}\text{-}\mathsf{SMDDH}_{\mathbb{G}_1}}(\mathcal{B}_3) + \\ & d\min(N - k_s, d)\log k_s \mathsf{Adv}_{\mathsf{SXDH}}(\mathcal{B}_4) + \frac{d(1+k_s)}{p}.\end{aligned}$$

Note that the most efficient, secure choice is $k_s = 2$ and then the largest security loss factor is $d\min(N - k_s, d) \leq d \cdot N$, which is at most the number of multiplicative gates in the circuit.

5.3 Efficiency

In the most efficient instantiation, the proof size is $(3d+2, d+2)$ group elements and naive verification requires to compute $3d$ pairings for the quadratic relations and $2(n_0+3d+4)$ for the linear part, and n_d exponentiations in $\mathbb{G}_1$ for the output. Using the "bilinear batching" techniques of Herold et al. [19] the number of pairings can be reduced to $n_0 + 3d + 4$ for the linear part. Since the input is known in $\mathbb{Z}_p$, n_0 pairings in this part can be replaced by n_0 exponentiations in $\mathbb{G}_T$. Finally, using standard batching techniques [6], the number of pairings for the quadratic part can be reduced to $d + 2$. As a result the total number of pairings required for verification is $4d + 6$, plus n_0 exponentiations in $\mathbb{G}_T$ and $O(n_0 + d + n_d)$ exponentiations in the source group.

In the instantiation which is secure under standard assumptions, the proof size is $(6d+3, 2d+3)$ group elements and naive verification requires to compute $6d$ pairings for the quadratic relations and $2(n_0 + 6d + 6)$ for the linear part, and using the same batching techniques the number of pairings required for verification is $8d + 9$.

5.4 Adding Zero-Knowledge

In this section we argue how to add zero-knowledge to the argument for correct arithmetic circuit evaluation of Sect. 5.2. The same discussion applies for the argument for boolean circuit satisfiability discussed in Sect. 6.1 for boolean circuits.

We have to distinguish two different situations. In the first one the input is public, and we can easily modify our proof so that it reveals nothing about the internal evaluation steps. When the input or part of the input must be secret, which is the most useful case, the circuit input cannot be part of the verifier's input, at least not in the clear. A natural idea is to let the prover commit to it. The problem is that our "knowledge transfer" idea requires the reduction in the soundness proof to know this secret input, which means that the commitment to the input must be extractable so that we can efficiently extend it to a vector of correct evaluations $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$. Even in a QA-NIZK setting where we can efficiently open the commitments, they are only F-extractable [3] (under falsifiable assumptions), which means that we can only extract in the source groups but not in $\mathbb{Z}_p$. This leaves us only with a couple of solutions, all of them unsatisfactory.

One of them is to commit to inputs bitwise and prove that this is done correctly. This is not acceptable in terms of concrete efficiency for arithmetic circuits, but it is a practical approach for boolean circuits.

The second one is to use a commitment to the input which is extractable under knowledge assumptions. Of course, then our construction is no longer secure under falsifiable assumptions, but it is interesting that it indicates a tradeoff in SNARK constructions: longer proof size and verification costs ($\Theta(d)$ group elements/ pairings, respectively) but weaker assumptions (only the input needs to be extracted and not the full witness).

In any case, we leave for future work to explore the possibilities of this or other mixed approaches (like using ROM based constructions for extracting the input). We now give the technical details on how to add zero-knowledge to our argument for correct circuit evaluation, distinguishing the two aforementioned situations.

Adding Zero-knowledge to Correct Evaluation of Middle Wires. This step is straightforward. The argument is changed so that $[\boldsymbol{L}]_1, [\boldsymbol{R}]_2, [\boldsymbol{O}]_1$ are not given in the clear, but instead the prover gives GS commitments [18] to each of its components. For the quadratic argument, it gives a GS Proof that the verification equation is satisfied, that is, for each i it proves in zk that the pairing product equation:

$$e([L_i]_1, [R_i]_2) - e([O_i]_1, [1]_2) = e([H_i]_1, [T]_2)$$

is satisfied, where $[L_i]_1, [R_i]_2, [O_i]_1, [H_i]_1$ are hidden committed values.

For the linear argument, it suffices to give a GS proof of satisfiability of the verification equation in Fig. 4. In its most efficient instantiation, the verification equation in Fig. 4 consists of 2 pairing product equations and hence the GS proof consists of 8 elements of each group. An alternative, more efficient approach (which requires only $(2, 2)$ group elements) for the linear argument proves that the vectors of committed elements are in a certain linear (bilateral) space. The idea is quite simple but details are a bit cumbersome, so we explain it in the full version of this work.

Hiding the input and output. Finally, we discuss how to use our results in a scenario where not only the middle wires should be hidden but also the input and the output. In this case the prover should commit to the input and the output with perfectly binding commitments $(\boldsymbol{c}_{\boldsymbol{x}}, \boldsymbol{d}_y)$.

The commitment to the input should be extractable. For instance, $\boldsymbol{c}_{\boldsymbol{x}}$ can be just the concatenation of GS commitments to the inputs provided the prover submits also a proof of knowledge of their opening (giving additional bitwise commitments and a proof that $\boldsymbol{c}_{\boldsymbol{x}}$ is of the right form, or a proof with knowledge assumptions or in the ROM). In any case, we require $\boldsymbol{c}_{\boldsymbol{x}}$ to be algebraic, that is, it should be possible to write it as $\boldsymbol{c}_{\boldsymbol{x}} = [\mathbf{E}]_1\boldsymbol{x} + [\mathbf{V}]_1\boldsymbol{r}$, where $\boldsymbol{r}$ is the vector of randomness and matrices $\mathbf{E}$, $\mathbf{V}$ are described in the commitment key (we can also allow $\boldsymbol{c}_{\boldsymbol{x}}$ to have components in both $\mathbb{G}_1, \mathbb{G}_2$, in which case $\mathbf{E}$ and $\mathbf{V}$ will be split). The only difference with the case where the commitment is public is that in the first n_0 rows of $\mathbf{M}$ the identity matrix should be replaced by $\mathbf{E}$ and an additional column of the form $\left(\mathbf{V}, \mathbf{0}\right)^\top$ should be added.

The prover should also give a GS proof that $\boldsymbol{d}_y$ opens to the same value as $[\boldsymbol{O}_d]_1$.

6 Boolean Circuits

We extend our results to any boolean circuit $\phi : \{0,1\}^{n_0} \to \{0,1\}^{n_d}$. The gates of ϕ are assumed to have fan-in two but otherwise they can be of any type (excluding non-interesting or trivial gate types). The construction relies on the characterization of these gates as quadratic functions of the inputs. We list below the 10 gate types allowed for the circuit ϕ, along with its expression as a quadratic function. The list of gates is taken from [1], which observe that the last remaining 6 gate types depend mostly on one input and are not used often.

$\mathsf{AND}(a,b,c)$: $ab = c$ $\mathsf{NAND}(a,b,c)$: $1 - ab = c$
$\mathsf{OR}(a,b,c)$: $1-(1-a)(1-b) = c$ $\mathsf{NOR}(a,b,c)$: $(1-a)(1-b) = c$
$\mathsf{XOR}(a,b,c)$: $b(1-a) + a(1-b) = c$ $\mathsf{XNOR}(a,b,c)$: $1 - a(1-b) - b(1-a) = c$
$\mathsf{G}_1(a,b,c) = (c = \overline{a} \wedge b)$: $(1-a)b = c$ $\mathsf{G}_2(a,b,c) = (c = \overline{\overline{a} \wedge b})$: $1-(1-a)b = c$
$\mathsf{G}_3(a,b,c) = (c = a \wedge \overline{b})$: $a(1-b) = c$ $\mathsf{G}_4(a,b,c) = (c = \overline{a \wedge \overline{b}})$: $1 - a(1-b) = c$.

From this characterization we slice the circuit into several quadratic and affine constraints similar to the arithmetic case. As before, we partition the set of gates $\mathcal{G}$ of a given circuit ϕ into different subsets $\mathcal{G}_i$ according to the depth, n_i is cardinal of the gates at level i and we assume that gates at each level are ordered in some way and they are denoted as $G_{i1}, \ldots, G_{in_i}$.

For each level i, we define variables C_{ij}, $j = 1, \ldots, n_i$ which will encode the output of gate j at level i. The gate G_{ij} will be correctly evaluated if $C_{ij} = G_{ij}(A_{ij}, B_{ij})$, where $A_{ij} = C_{k_L \ell_L}$ and $B_{ij} = C_{k_R \ell_R}$ for some indexes $0 \leq k_L, k_R < i$, $1 \leq \ell_L \leq n_{k_L}$ and $1 \leq \ell_R \leq n_{k_R}$, which depend on i, j and which are specified by the circuit description. That is, the left wire of G_{ij} should be the output of the ℓ_Lth gate at level k_L and the right wire the output of the ℓ_Rth gate at level k_R.

Lemma 7. *Let $\phi : \{0,1\}^{n_0} \to \{0,1\}^{n_d}$, be a circuit of multiplicative depth d with n_i gates at level i. There exist*

a) variables C_{ij}, $i = 0, \ldots, d$, $j = 1, \ldots, n_i$,
b) variables A_{ij}, B_{ij}, $i = 1, \ldots, d$, $j = 1, \ldots, n_i$,
c) constants $f_{ijk\ell}, g_{ijk\ell} \in \{0,1\}$, $i = 1, \ldots, d$, $k = 0, \ldots, i-1$, $j = 1, \ldots, n_i$, $\ell = 1, \ldots, n_k$,
d) constants $\beta_{ij}, \gamma_{ij}, \epsilon_{ij}, \delta_{ij} \in \mathbb{Z}_p$, $i = 1, \ldots, d$, $j = 1, \ldots, n_i$, which depend on the type of gate G_{ij},

such that, for every $(x_1, \ldots, x_{n_0}) \in \{0,1\}^{n_0}$, if we set $C_{0,j} = x_j$, for all $j = 1, \ldots, n_0$, then $\phi(\boldsymbol{x}) = \boldsymbol{y}$ and A_{ij}, C_{ij} are evaluated to the left and output of the jth gate at level i, if and only if the following equations are satisfied:

1. *(Quadratic constraints). For each $i = 1, \ldots, d$, for all $j = 1, \ldots, n_i$,*

$$C_{ij} = A_{ij}B_{ij} + A_{ij}\beta_{ij} + B_{ij}\gamma_{ij} + \epsilon_{ij}, \tag{9}$$

2. *(Affine constraints) $A_{ij} = \sum_{k=0}^{i-1}\sum_{\ell=1}^{n_k} f_{ijk\ell}C_{k\ell}$ and $B_{ij} = \sum_{k=0}^{i-1}\sum_{\ell=1}^{n_k} g_{ijk\ell}C_{k\ell}$.*
3. *(Correct Output) For all $j = 1, \ldots, n_d$, $C_{dj} = y_j$.*

Proof. For the $(i,j)th$ circuit gate, a description of the circuit ϕ specifies the gate type and indexes $(k_{i,j,L}, \ell_{i,j,L})$ which indicate the left and right wire. Therefore, from the quadratic expression of boolean gates for boolean circuit satisfiability, correct evaluation of G_{ij} is expressed as:

$$C_{ij} = C_{k_{i,j,L},\ell_{i,j,L}}C_{k_{i,j,R},\ell_{i,j,R}}\alpha_{ij} + C_{k_{i,j,L},\ell_{i,j,L}}\beta_{ij} + C_{k_{i,j,R},\ell_{i,j,R}}\hat{\gamma}_{ij} + \epsilon_{ij},$$

for some $\alpha_{ij}, \beta_{ij}, \hat{\gamma}_{ij}, \epsilon_{ij} \in \mathbb{Z}$ which depend on the gate type. This can be rewritten as an equation over $\mathbb{Z}_p$ as:

$$C_{ij} = C_{k_{i,j,L},\ell_{i,j,L}}(C_{k_{i,j,R},\ell_{i,j,R}}\alpha_{ij}) + C_{k_{i,j,L},\ell_{i,j,L}}\beta_{ij} + (C_{k_{i,j,R},\ell_{i,j,R}}\alpha_{ij})(\alpha_{ij}^{-1}\hat{\gamma}_{ij}) + \epsilon_{ij}. \tag{10}$$

For any (i,j) we define the constant $f_{ijk\ell}$ and $g_{ijk\ell}$ to be 0 everywhere except for $f_{ijk_{i,j,L}\ell_{i,j,L}} = 1$ and $g_{ijk_{i,j,R}\ell_{i,j,R}} = \alpha_{ij}$. Therefore, if $A_{ij} = \sum_{k=0}^{i-1}\sum_{\ell=1}^{n_k} f_{ijk\ell}C_{k\ell} = C_{k_{i,j,L},\ell_{i,j,L}}$ and $B_{ij} = \sum_{k=0}^{i-1}\sum_{\ell=1}^{n_k} g_{ijk\ell}C_{k\ell} = C_{k_{i,j,R},\ell_{i,j,R}}$ and equation (10) which expresses correct evaluation of gate (i,j) can be rewritten as:

$$C_{ij} = A_{ij}B_{ij} + A_{ij}\beta_{ij} + B_{ij}\gamma_{ij} + \epsilon_{ij}, \tag{11}$$

where $\gamma_{ij} = \alpha_{ij}^{-1}\hat{\gamma}_{ij}$.

Obviously, this implies that if $c_{0,j} = x_j$, and the linear constraints are satisfied, then the rest of the output wires are also consistent with x_j and we conclude that $c_{n_d,j}$ is the output corresponding to this input. Therefore, if $c_{n_d,j} = y_j$, we can conclude that $\phi(\boldsymbol{x}) = \boldsymbol{y}$.

To achieve succinct ness, quadratic equations which encode correct gate evaluation are represented as a divisibility relation with the usual polynomial aggregation technique.

Lemma 8. *Let $\mathcal{R} \subset \mathbb{Z}_p$ be a set of cardinal N and let $\lambda_j(X)$ be the associated Lagrangian polynomials and $t(X)$ the polynomial whose roots are the elements of $\mathcal{R}$. Let $\phi : \{0,1\}^{n_0} \to \{0,1\}^{n_d}$, be any circuit such that $N = \max_{i=1,\ldots,d} n_i$. There exist some unique polynomials $u_{L,i}(X), u_{R,i}(X), u_{0,i}(X)$ of degree at most $N-1$ which are efficiently computable from the circuit description and such that for any tuple $(\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i) \in (\{0,1\}^{n_i})^3$, if*

$$\ell_i(X) = \sum_{j=1}^{n_i} a_j \lambda_j(X), \qquad r_i(X) = \sum_{j=1}^{n_i} b_j \lambda_j(X), \qquad o_i(X) = \sum_{j=1}^{n_i} c_j \lambda_j(X),$$

it holds that $\boldsymbol{a}_i, vecc_i$ are consistent assignments to the left and output values of gates at level i if and only if $t(X)$ divides $p_i(X)$, where

$$p_i(X) = \ell_i(X) r_i(X) + \ell_i(X) u_{L,i}(X) + r(X) u_{R,i}(X) + u_{0,i}(X) - o_i(X).$$

Proof. The proof is a direct consequence of Lemma 7. Indeed, it suffices to define $u_{L,i}(X), u_{R,i}(X), u_{0,i}(X)$ to take the values $u_{L,i}(r_j) = \beta_{ij}$, $u_{R,i}(r_j) = \gamma_{ij}$ and $u_{0,i}(r_j) = \epsilon_{ij}$ for $j = 1, \ldots, n_i$ and 0 for $j = n_i + 1, \ldots, N$. Therefore, $p_i(r_j) = a_{ij} b_{ij} + a_{ij} \beta_{ij} + b_{ij} \gamma_{ij} + \epsilon_{ij} - c_{ij}$. This proves that if equation (11) is satisfied then $p_i(X)$ is divisible by $t(X)$, since it is 0 in all of its roots. Finally, the polynomials $u_{L,i}(X), u_{R,i}(X), u_{0,i}(X)$ can be efficiently computed from the circuit description, as they depend only on N and the type of each gate.

6.1 A New Argument for Correct Boolean Circuit Evaluation

From Lemma 7, we can design an argument for boolean circuit satisfiability based on falsifiable assumptions, similar as in Sect. 5. The argument is based on a quadratic and a linear "knowledge transfer" subarguments. The value $[R_i]_2$ is now defined as $[\boldsymbol{R}_i]_2 = \sum_{j=1}^{n_i} \alpha_{ij} b_{ij} \lambda_j(\boldsymbol{s})$. The linear argument is identical to the arithmetic case.

For the quadratic argument, now the prover needs to show (aggregating the proof at each level i for $j = 1, \ldots, n_i$) that the quadratic equations $C_{ij} = A_{ij} B_{ij} + A_{ij} \beta_{ij} + B_{ij} \gamma_{ij} + \epsilon_{ij}$ are satisfied, whereas before the equations were $C_{ij} = A_{ij} B_{ij}$. However, the security proof is almost identical to the arithmetic case.

Indeed, the verification equation of the quadratic argument is adapted to the new equation type, i.e. For each level $i = 1, \ldots, d$, and each $j = 1, \ldots, k_s$ given commitments $[L_{ij}]_1, [R_{ij}]_2, [O_{ij}]_1$, and some value $[H_{ij}]_1$ the quadratic argument checks if

$$e([L_{ij}]_1, [R_{ij}]_2) + e([L_{ij}]_1, [u_{L,i}(s_j)]_2) + e([u_{R,i}(s_j)]_1, [R_{ij}]_2) + e([u_{0,i}(s_j)]_1, [1]_2) \\ - e([O_{ij}]_1, [1]_2) = e([H_{ij}]_1, [T]_2),$$

where $u_{L,i}(X), u_{R,i}(X), u_{0,i}(X)$ are the polynomials associated to the gate constants at level i. To prove soundness, given an opening of $[L_{ij}]_1$ and $[R_{ij}]_2$ which is not consistent with $[O_{ij}]$, it suffices to compute $[O'_{ij}]_1, [H'_{ij}]_1$ consistent with these openings and subtract the two verification equations to find a solution to the $\mathcal{R}$-Rational Strong Diffie-Hellman Assumption.

Zero-Knowledge. The argument can be made zero-knowledge for the middle wires by proving with the GS proof system that the argument for correct circuit evaluation is satisfied, as discussed in Sect. 5.4 for the arithmetic case. The input can also be hidden provided it is encrypted with an extractable commitment. In the boolean case this can be done in a relatively efficient way, for example under the DDH Assyumption with GS commitments. The cost of giving the committed secret inputs and a proof that they open to $\{0,1\}$ using the GS proof system is $(6(n_0 - n_{pub}), 6(n_0 - n_{pub}))$ group elements. It can be reduced to $(2(n_0 - n_{pub}) + 10, 10)$ group elements under standard assumptions using the results of González and Ràfols [14], but at the price of having a crs quadratic in n_0 and to $(2n_0 + 4, 6)$ with a linear crs under a non-standard (falsifiable) $(n_0 - n_{pub})$-assumption similar to the q-Target Strong Diffie-Hellman Assumption using the results of Daza et al. [2].

References

1. G. Danezis, C. Fournet, J. Groth, and M. Kohlweiss. Square span programs with applications to succinct NIZK arguments. In P. Sarkar and T. Iwata, editors, *ASIACRYPT 2014, Part I*, volume 8873 of *LNCS*, pages 532–550. Springer, Heidelberg, Dec. 2014. 2, 25
2. V. Daza, A. González, Z. Pindado, C. Ràfols, and J. Silva. Shorter quadratic QA-NIZK proofs. In D. Lin and K. Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 314–343. Springer, Heidelberg, Apr. 2019. 3, 28
3. A. Escala and J. Groth. Fine-tuning Groth-Sahai proofs. In H. Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 630–649. Springer, Heidelberg, Mar. 2014. 24
4. A. Escala, G. Herold, E. Kiltz, C. Ràfols, and J. L. Villar. An algebraic framework for Diffie-Hellman assumptions. *Journal of Cryptology*, 30(1):242–288, Jan. 2017. 3, 5
5. P. Fauzi, H. Lipmaa, J. Siim, and M. Zajac. An efficient pairing-based shuffle argument. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 97–127. Springer, Heidelberg, Dec. 2017. 3, 5, 14, 15
6. A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen. Practical short signature batch verification. In M. Fischlin, editor, *CT-RSA 2009*, volume 5473 of *LNCS*, pages 309–324. Springer, Heidelberg, Apr. 2009. 23
7. R. Gennaro, C. Gentry, and B. Parno. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In T. Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 465–482. Springer, Heidelberg, Aug. 2010. 7
8. R. Gennaro, C. Gentry, B. Parno, and M. Raykova. Quadratic span programs and succinct NIZKs without PCPs. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Heidelberg, May 2013. 2, 4, 7, 11
9. C. Gentry. Fully homomorphic encryption using ideal lattices. In M. Mitzenmacher, editor, *41st ACM STOC*, pages 169–178. ACM Press, May / June 2009. 6
10. C. Gentry, J. Groth, Y. Ishai, C. Peikert, A. Sahai, and A. D. Smith. Using fully homomorphic hybrid encryption to minimize non-interative zero-knowledge proofs. *Journal of Cryptology*, 28(4):820–843, Oct. 2015. 6

11. C. Gentry and D. Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In L. Fortnow and S. P. Vadhan, editors, *43rd ACM STOC*, pages 99–108. ACM Press, June 2011. 2
12. E. Ghadafi and J. Groth. Towards a classification of non-interactive computational assumptions in cyclic groups. In T. Takagi and T. Peyrin, editors, *ASIACRYPT 2017, Part II*, volume 10625 of *LNCS*, pages 66–96. Springer, Heidelberg, Dec. 2017. 3, 9
13. S. Goldwasser, Y. T. Kalai, and G. N. Rothblum. Delegating computation: interactive proofs for muggles. In R. E. Ladner and C. Dwork, editors, *40th ACM STOC*, pages 113–122. ACM Press, May 2008. 7
14. A. González, A. Hevia, and C. Ràfols. QA-NIZK arguments in asymmetric groups: New tools and new constructions. In T. Iwata and J. H. Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 605–629. Springer, Heidelberg, Nov. / Dec. 2015. 3, 4, 6, 9, 18, 19, 28
15. J. Groth. Short pairing-based non-interactive zero-knowledge arguments. In M. Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Heidelberg, Dec. 2010. 2
16. J. Groth. On the size of pairing-based non-interactive arguments. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016. 2, 7
17. J. Groth, R. Ostrovsky, and A. Sahai. New techniques for noninteractive zero-knowledge. *J. ACM*, 59(3):11, 2012. 6, 7
18. J. Groth and A. Sahai. Efficient noninteractive proof systems for bilinear groups. *SIAM J. Comput.*, 41(5):1193–1232, 2012. 3, 6, 24
19. G. Herold, M. Hoffmann, M. Klooß, C. Ràfols, and A. Rupp. New techniques for structural batch verification in bilinear groups with applications to groth-sahai proofs. In B. M. Thuraisingham, D. Evans, T. Malkin, and D. Xu, editors, *ACM CCS 2017*, pages 1547–1564. ACM Press, Oct. / Nov. 2017. 23
20. C. S. Jutla and A. Roy. Shorter quasi-adaptive NIZK proofs for linear subspaces. In K. Sako and P. Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Heidelberg, Dec. 2013. 4, 6
21. C. S. Jutla and A. Roy. Switching lemma for bilinear tests and constant-size NIZK proofs for linear subspaces. In J. A. Garay and R. Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 295–312. Springer, Heidelberg, Aug. 2014. 4, 6, 14
22. Y. Kalai, O. Paneth, and L. Yang. On publicly verifiable delegation from standard assumptions. Cryptology ePrint Archive, Report 2018/776, 2018. `https://eprint.iacr.org/2018/776`. 7
23. Y. T. Kalai, R. Raz, and R. D. Rothblum. How to delegate computations: the power of no-signaling proofs. In D. B. Shmoys, editor, *46th ACM STOC*, pages 485–494. ACM Press, May / June 2014. 7
24. S. Katsumata, R. Nishimaki, S. Yamada, and T. Yamakawa. Exploring constructions of compact NIZKs from various assumptions. In A. Boldyreva and D. Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 639–669. Springer, Heidelberg, Aug. 2019. 6
25. E. Kiltz and H. Wee. Quasi-adaptive NIZK for linear subspaces revisited. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, Apr. 2015. 3, 4, 5, 6, 14, 17, 18
26. B. Libert, T. Peters, M. Joye, and M. Yung. Non-malleability from malleability: Simulation-sound quasi-adaptive NIZK proofs and CCA2-secure encryption

from homomorphic signatures. In P. Q. Nguyen and E. Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 514–532. Springer, Heidelberg, May 2014. 14, 17

27. B. Libert, T. Peters, and M. Yung. Short group signatures via structure-preserving signatures: Standard model security from simple assumptions. In R. Gennaro and M. J. B. Robshaw, editors, *CRYPTO 2015, Part II*, volume 9216 of *LNCS*, pages 296–316. Springer, Heidelberg, Aug. 2015. 4, 6
28. H. Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In R. Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189. Springer, Heidelberg, Mar. 2012. 2
29. C. Lund, L. Fortnow, H. J. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. In *31st FOCS*, pages 2–10. IEEE Computer Society Press, Oct. 1990. 7
30. M. Maller, M. Kohlweiss, S. Bowe, and S. Meiklejohn. Sonic: Zero-knowledge snarks from linear-size universal and updateable structured reference string. Cryptology ePrint Archive, Report 2019/099, 2019. `http://eprint.iacr.org/2019/099`. 7
31. P. Morillo, C. Ràfols, and J. L. Villar. The kernel matrix Diffie-Hellman assumption. In J. H. Cheon and T. Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 729–758. Springer, Heidelberg, Dec. 2016. 9
32. M. Naor. On cryptographic assumptions and challenges (invited talk). In D. Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, Aug. 2003. 2
33. O. Paneth and G. N. Rothblum. On zero-testable homomorphic encryption and publicly verifiable non-interactive arguments. In Y. Kalai and L. Reyzin, editors, *TCC 2017, Part II*, volume 10678 of *LNCS*, pages 283–315. Springer, Heidelberg, Nov. 2017. 7
34. B. Parno, J. Howell, C. Gentry, and M. Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society Press, May 2013. 7
35. C. Peikert and S. Shiehian. Noninteractive zero knowledge for np from (plain) learning with errors. Cryptology ePrint Archive, Report 2019/158, 2019. `https://eprint.iacr.org/2019/158`. 6