On the efficiency of pairing-based proofs under the d-PKE

Ariel Gabizon[*]

Abstract

We investigate the minimal number of group elements and prover running time in a zk-SNARK when using only a symmetric “linear” knowledge assumption, like the d-Power Knowledge of Exponent assumption, rather than a “quadratic” one as implicitly happens in the most efficient known construction by Groth [Gro16]. The proofs of [Gro16] contain only 3 group elements. We present 4 element proofs for quadratic arithmetic programs/rank 1 constraint systems under the d-PKE with very similar prover running time to [Gro16]. Central to our construction is a simple lemma for “batching” knowledge checks, which allows us to save one proof element.

1 Introduction

The most efficient known proving systems with succinct proofs rely on very strong cyrptographic assumptions, sometimes called non-falsifiable [Nao03]. This was shown to be necessary in a sense [GW11]. More specifically, the type of non-falsifiable assumptions used are refered to as *knowledge assumptions*. Rather than being assumptions of what is hard for an adversary to do, they are of the form: *if* the adversary succeeded in a certain task, *he must have done it in a certain way*. More concretely, the assumptions are typically about answers to certain challenges being linear combinations of CRS elements, which translates to an extractor being able to output the coefficients of this combination.

The aim of this work is to try to minimize the type of knowledge assumption made, while preserving efficiency as much as possible.

1.1 On linear and quadratic knowledge assumptions

We present a framework for instantiating different knowledge assumptions to motivate our result. The definitions in this section are presented at a semi-formal level, as besides motivating the result, they are not needed for formally deriving or stating it.

In the following, for $a \in \mathbb{F}$, $[a]$ denotes the group element $a \cdot g \in \mathbb{G}$ where $\mathbb{F}$ is a finite field of prime order and $\mathbb{G}$ an additive group of equal order. (See Section 1.3 for more details on notation and terminology.)

Let us recall the d-Power Knowledge of Exponent assumption (d-PKE) originally introduced by Groth [Gro10], and central to many SNARK constructions, e.g. [GGPR13, PHGR16]: The adversary $\mathcal{A}$ receives a set of encoded elements $\left\{\left[\tau^i\right], \left[\alpha\tau^i\right]\right\}_{i\in[0..d]}$, together with other elements independent of α. $\mathcal{A}$ is then given the challenge of producing another pair of ‘ratio’ α; i.e. a pair

[*] Part of this work was done while being supported by the Zcash Company.

of the form $([c], [c']) = ([c], \alpha \cdot [c])$. Note that a natural way to answer the challenge is by taking $[c]$ to be some linear combination of the elements $\{[\tau^i]\}$ and by taking $[c']$ to be the corresponding combination of the elements $\{[\alpha\tau^i]\}$. That is, taking

$$[c] := \sum_{i=0}^{d} a_i \cdot [\tau^i], [c'] := \sum_{i=0}^{d} a_i \cdot [\alpha\tau^i],$$

for some $\mathbf{a} = (a_0, \ldots, a_d) \in \mathbb{F}^{d+1}$. The d-PKE states that this is the *only way* for $\mathcal{A}$ to succeed with non-negligible probability. "Only way" is then formalized by saying that whenever $\mathcal{A}$ succeeds in the challenge, another algorithm E, a.k.a. the extractor, will succeed in outputting the corresponding $\mathbf{a}$.

Let us abstract what is going on in the d-PKE, so that we may generalize it. We had a *challenge equation*

$$Y_2 = \alpha \cdot Y_1.$$

$\mathcal{A}$ was challenged to find group elements $[c]$,$[c']$ such that encoded elements c, c' satisfy the equation. $\mathcal{A}$ is given a *challenge set* of elements in order to aid him in completing the challenge. In the d-PKE the challenge set is always of the form $\{[\tau^i], [\alpha\tau^i]\}_{i\in[0..d]}$ together with elements that are independent of α.

With this terminology, it is not hard to see the d-PKE is equivalent to the following slightly more abstract phrasing: The only way to satisfy the challenge equation is by taking $[c], [c']$ to be linear combinations of the challenge elements such that the equation holds as a polynomial identity in τ, α (and possibly other variables appearing in the auxiliary information independent of α).

Now, given this terminology and phrasing of the assumption, it is immediate to see how it generalizes. Instead of looking at the equation $Y_2 = \alpha \cdot Y_1$, we could look at any equation.

For example, a multi-variate linear equation $Y_t = \alpha_1 \cdot Y_1 + \ldots + \alpha_{t-1} \cdot Y_{t-1}$. Lemma 2.3 will roughly show that this equation does not lead to a stronger assumption than the two variable original version.

Naturally, we could also look at higher degree challenge equations. When our proofs are pairing-based, as the equations are typically derived from verifier checks, there is no point in looking at degree larger than two. Assuming $\mathcal{A}$ is a generic group adversary as in the security proof of Groth [Gro16] implies[1] making the assumption for any degree d equation and challenge set when $d = \mathrm{poly}(\lambda)$. In particular, instead of using the generic group model, the security proof of [Gro16] can be done by making such an assumption for some *quadratic* equation.

Arguably, assumptions involving quadratic equations are stronger than assumptions regarding linear equations, which is why it is of value to maximize SNARK efficiency while restricting oneself to linear assumptions like the d-PKE - this being the purpose of our work.

Symmetric vs asymmetric assumptions Groth and Maller [GM17] cleverly notice that when using an asymmetric pairing even the trivial equation $Y_2 = Y_1$ can lead to a plausible assumption when requiring c and c' to be encoded in the distinct source groups; and that using square arithmetic programs inspired by Danezis et. al [DFGK14], one can get a 3 element proof as in [Gro16] under only the linear assumption corresponding to this equation (with a "bonus" of obtaining simulation

[1] This is strictly true when the auxiliary data is "low-degree" in the sense described in Definition 2.1 and Remark 2.2

extractability which was the focus of that work). However, the use of SAPs rather than QAPs increases field and group operations significantly when starting with an arithmetic circuit; and arguably a symmetric linear assumption is better than the non-symmetric one used by [GM17] - which can roughly be seen as assuming there is no efficiently computable isomorphism between the two pairing source groups, even when requiring correct computation on only a polynomially small fraction of inputs.

1.2 Relation to previous work

As alluded to above, the most relevant works for comparison are [Gro16, GM17]. We also mention that our security proof is very much inspired by that of [GM17]. We summarize the tradoffs between the three works. Suppose we start with an arithmetic circuit with n multiplication gates and m wires. We think of $\ell < m$ of the wires as public (usually these will all be input and output wires of the circuit), and are interested in a zk-SNARK showing that given a value x of the public wires, the prover knows an assignment to the other $m - \ell$ wires consistent with the circuit computation. Below $E_1(E_2)$ denotes exponentations in $\mathbb{G}_1(\mathbb{G}_2)$, P means pairings.

Pairing based SNARKs are almost always instantiated using asymmetric pairings, where operations in the second source group $\mathbb{G}_2$ are considerably more expensive than in the first.

The main point here is that we preserve the amount of $\mathbb{G}_2$ prover operations from [Gro16], rather than doubling it as in [GM17], while adding roughly only n $\mathbb{G}_1$ operations.

[Gro16]: ***Size:*** 2 $\mathbb{G}_1$, 1 $\mathbb{G}_2$. ***Prover operations:*** $m + 3n - \ell + 3$ E_1, $n + 1$ E_2. ***Verifier operations:*** ℓ E_1, 3 P. ***Knowledge assumption:*** Quadratic symmetric.

[GM17]: ***Size:*** 2 $\mathbb{G}_1$, 1 $\mathbb{G}_2$. ***Prover operations:*** $m+4n-\ell$ E_1, $2n$ E_2. ***Verifier computation:*** ℓ E_1, 5 P, ***Knowledge assumption:*** Linear asymmetric.

This work: ***Size:*** 3 $\mathbb{G}_1$, 1 $\mathbb{G}_2$. ***Prover operations:*** $m + 2n + \min(2n, m + 2) - \ell + 2$ E_1, n E_2. ***Verifier computation:*** ℓ E_1, 5 P. ***Knowledge assumption:*** Linear symmetric.

1.3 Terminology and conventions

We assume we are given a finite field $\mathbb{F}$ and group $\mathbb{G}$ both of the same prime order r, together with a generator $g \in \mathbb{G}^*$. For $x \in \mathbb{F}$ we denote $[x] := x \cdot g$ and refer to $[x]$ as an *encoding of* x. For a set or vector T of elements of $\mathbb{F}$, we refer by $[T]$ to the corresponding set or vector of element encodings; e.g. $[(a_1, \ldots, a_t)] := ([a_1], \ldots, [a_t])$.

We assume our common reference strings are always of the following form. We have a fixed map $f : \mathbb{F}^t \to \mathbb{F}^M$ where for each $i \in [t]$ $f_i(X)$ is a rational function of total degree at most d in both numerator and denumerator, and the common reference string is of the form $[f(\mathbf{x})]$ for uniform $\mathbf{x} \in \mathbb{F}^t$

We assume all objects and algorithms are dependent on an implicit integer parameter λ. For example, when we refer to a field $\mathbb{F}$, we implicitly mean an infinite sequence of fields $\mathbb{F}(\lambda)$ indexed by λ. When we refer to a party $\mathcal{A}$ as *efficient* we mean a circuit of size poly(λ). When we say a function is *efficiently computable* we mean it is computable by a uniform algorithm of running time poly(λ).

Moreover, we assume existence of an efficient group generator $\mathscr{G}$, that given λ outputs representations of groups $\mathbb{G}, \mathbb{G}_t$ and a finite field $\mathbb{F}$ all of prime order $r \geq 2^\lambda$, and a uniformly chosen generator $g \in \mathbb{G}$. We also assume the existence of an efficiently computable non-degenerate bi-linear pairing $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_t$. Whenever stating a theorem or assumption, we assume the involved parties have access to the output of $\mathscr{G}$. So, when stating a cryptographic assumption, the assumption implicitly depends on the group generator used.

2 Cryptographic assumptions

We formally define the d-PKE that was described in the previous section.

Definition 2.1 (d-Power Knowledge of Exponent Assumption (d-PKE))**.** *For any efficient $\mathcal{A}$ there exists an efficient E such that the following holds. Fix a constant t and $M = \mathrm{poly}(\lambda)$, and an efficiently computable degree d rational map $S : \mathbb{F}^{t+1} \to \mathbb{F}^M$. Consider the following experiment.*

$\tau, \alpha \in \mathbb{F}, \mathbf{x} \in \mathbb{F}^t$ are chosen uniformly. We denote $V := (1, \tau, \ldots, \tau^d, \alpha, \alpha\tau, \ldots, \alpha\tau^d)$ Then $\mathcal{A}$ is given as input $([V], S(\tau, \mathbf{x}))$ and outputs a pair $([c], [c'])$ of $\mathbb{G}$ elements, which he "hopes" is of the form $([c], [\alpha c])$. E, given the same input, outputs a polynomial $A \in \mathbb{F}[X]$ of degree at most d. The probability that both

1. *$\mathcal{A}$ "succeeded", i.e., $c' = \alpha \cdot c$. But,*
2. *E "failed", i.e., $c \neq A(\tau)$.*

is $\mathsf{negl}(\lambda)$*.*

Remark 2.2. *Typically the PKE assumption is defined with arbitrary auxiliary information rather than a rational map. Our definition is weaker and corresponds to imposing the auxiliary information to be the encoded output of a low degree rational map on a uniform input, and suffices for the security proofs of [GGPR13, PHGR16]. This increases the chance our auxiliary information is "benign" and our assumption does not contradict indistinguishability obfuscation [BCPR13].*

We show that the d-PKE can be used to "batch" knowledge checks.

Lemma 2.3. *Assuming the d-PKE the following holds. Fix $k = \mathrm{poly}(\lambda)$, a constant t and an efficiently computable degree d rational map $S : \mathbb{F}^{t+1} \to \mathbb{F}^M$. Fix any $i \in [k]$. For any efficient $\mathcal{A}$ there exists an efficient E such that the following holds. Consider the following experiment. $\alpha_1, \ldots \alpha_k, \tau \in \mathbb{F}$ and $\mathbf{x} \in \mathbb{F}^t$ are chosen uniformly. $\mathcal{A}$ is given as input $[S(\tau, \mathbf{x})]$ and $\left\{ \left[\alpha_j \cdot \tau^\ell\right] \right\}_{j \in [k], \ell \in [0..d]}$ and outputs a sequence of elements $([a_1], \ldots, [a_k], [b])$ in $\mathbb{G}$. E, given the same input as $\mathcal{A}$ together with the randomness of $\mathcal{A}$ and $\{\alpha_j\}_{j \in [k] \setminus \{i\}}$, outputs $A(X) \in \mathbb{F}[X]$ of degree at most d such that the probability that both*

1. *$\mathcal{A}$ "succeeded", i.e., $b = \sum_{j=1}^{k} \alpha_j \cdot a_j$. But,*
2. *E "failed", i.e., $a_i \neq [A(\tau)]$.*

is $\mathsf{negl}(\lambda)$*.*

Proof. Fix $k, t, S, \mathcal{A}$ and $i \in [k]$ as in the lemma. Assuming the d-PKE we aim to construct E satisfying the lemma statement. Consider the following efficient $\mathcal{A}^*$, that on input $[S(\tau, \mathbf{x})] \cup \left\{ \left[\alpha_i \tau^\ell\right] \right\}_{\ell \in [0..d]}$ samples random $\{\alpha_\ell \in \mathbb{F}^*\}_{\ell \in [k] \setminus \{i\}}$, computes $\left\{ \left[\alpha_j \tau^\ell\right] \right\}_{j \in [k] \setminus \{i\}, \ell \in [0..d]}$ and invokes $\mathcal{A}$ with uniformly chosen randomness $\mathsf{rand}_{\mathcal{A}}$ on $[S(\tau, \mathbf{x})] \cup \left\{ \left[\alpha_j \cdot \tau^\ell\right] \right\}_{j \in [k], \ell \in [0..d]}$. When $\mathcal{A}$ returns $([a_1], \ldots, [a_k], [b])$, $\mathcal{A}^*$ returns $([a_i], [b'])$ where

$$\left[b'\right] = [b] - \sum_{j \in [k] \setminus \{i\}} \alpha_j \cdot [a_j].$$

Note that $\mathcal{A}$ succeeds exactly when $\mathcal{A}^*$ succeeds in the sense that

$$\sum_{j \in [k]} \alpha_j \cdot a_j = b \Leftrightarrow \alpha_i \cdot a_i = b'.$$

Let E' be the extractor guaranteed to exist for $\mathcal{A}^*, S$ from the d-PKE. Note that the input for E' is $[S(\tau, \mathbf{x})] \cup \left\{ \left[\alpha_i \tau^\ell\right] \right\}_{\ell \in [0..d]}$ together with the inner randomness of $\mathcal{A}^*$ which is $\mathsf{rand}_{\mathcal{A}}, \{\alpha_\ell\}_{\ell \in [k] \setminus \{i\}}$. Given this input the probability that $\mathcal{A}^*$ succeeds in outputting a pair $([a_i], [b'])$ with $b' = \alpha_i \cdot a_i$, and E doesn't output A of degree at most d with $A(\tau) = a_i$ is $\mathsf{negl}(\lambda)$. Now define E to be the (identical) extractor that given $\mathsf{rand}_{\mathcal{A}}, [S(\tau, \mathbf{x})] \cup \left\{ \left[\alpha_i \tau^\ell\right] \right\}_{\ell \in [0..d]}, \{\alpha_\ell\}_{\ell \in [k] \setminus \{i\}}$, simply returns the output A of E' on the same input. □

The following assumption generalizes the d-SDH and d-PDH used in [GGPR13, PHGR16], and is very similiar to computational polynomial assumption from [GM17], except that it also allows rational functions.

The d-PDH for example, says you should not be able to output $\left[\tau^{d+1}\right]$ after seeing encodings of smaller powers of τ. The generalization here is that we assume an adversary, after seeing encoded evaluations of multi-variate rational functions, can only knowingly output an encoding of a rational function that is in the span of those it has seen.

Definition 2.4 ((t,d)-SPAN)**.** *Fix integers t, d. Fix an efficiently computable degree d rational map $S : \mathbb{F}^t \to \mathbb{F}^M$. Let V be the $\mathbb{F}$-subspace of $\mathbb{F}(X_1, \ldots, X_t)$ spanned by the output coordinates of S; i.e. $V := \mathsf{span}(\{S_i\}_{i \in [M]})$. Fix any efficient $\mathcal{A}$, and consider the following game: Uniform $\mathbf{x} \in \mathbb{F}^t$ is sampled and $\mathcal{A}$ is given $[S(\mathbf{x})]$. Then the probability that $\mathcal{A}$ outputs $p, q \in \mathbb{F}(X)$ and $h \in \mathbb{G}$ such that*

1. $\deg(p), \deg(q) \leq d$.
2. $p/q \notin V$.
3. $h = [p(\tau)/q(\tau)]$

is $\mathsf{negl}(\lambda)$

3 SNARK definitions

We formally define zk-SNARKs. We make a slightly non-conventional definition of knowledge soundness, where we allow the knowledge extractor access to part of the CRS trapdoor.

Definition 3.1. *An zk-SNARK $\mathscr{S}$ (zero-knowledge Succinct Non-interactive Argument of Knowledge) for a relation $\mathcal{R}$ consists of the following four possibly randomized algorithms.*

1. Gen *outputting a pair of trapdoors* $(r_{\text{ext}}, r_{\text{sim}})$ *and common reference string* σ.
2. P *that takes as input* σ *and* $(x, \omega) \in \mathcal{R}$ *and outputs* π.
3. V *that takes as input a common reference string* σ, *an input* x, *and a proof* π, *and outputs a value in* $\{\mathsf{acc}, \mathsf{rej}\}$.
4. P^{sim} *taking as input* x, *and trapdoor* r_{sim} *and outputting* π. *(It will be convenient to think of* P^{sim} *as returning* (x, π)*.)*

The quadruple of algorithms $\mathscr{S} = (\mathsf{Gen}, \mathrm{P}, \mathsf{V}, \mathrm{P}^{\text{sim}})$ *is a zk-SNARK for* $\mathcal{R}$ *if it satisfies*

1. ***Completeness:*** *For any common reference string* σ *output by* Gen, *and any* $(x, \omega) \in \mathcal{R}$, *if* $\pi = \mathrm{P}(\sigma, x, \omega)$*; then* $\mathsf{V}(\sigma, x, \pi) = \mathsf{acc}$ *with probability one.*
2. ***Statistical Zero-Knowledge:*** *For any output* (r, σ) *of* Gen *and* $(x, \omega) \in \mathcal{R}$, *the distribution*[2] *of* $\mathrm{P}^{\text{sim}}(r, x)$ *is* $\mathsf{negl}(\lambda)$*-close to that of* $\mathrm{P}(\sigma, x, \omega)$.
3. ***Knowledge Soundness:***

   For any efficient adversary $\mathcal{A}$, *there exists an efficient* E *such that the following holds: Suppose that* $\mathcal{A}$, *given* σ, *outputs a pair* (x, π) *and* E *given* $\mathsf{rand}_{\mathcal{A}}$ *and* r_{ext}, *outputs* ω. *The probability, over the randomness of* $\mathcal{A}$ *and that of* Gen *while outputting* $((r_{\text{ext}}, r_{\text{sim}}), \sigma)$, *that*

   - $\mathcal{A}$ *"wins":* $\mathsf{V}(x, \pi, \sigma) = \mathsf{acc}$, *and*
   - E *"loses":* $(x, \omega) \notin \mathcal{R}$

   is $\mathsf{negl}(\lambda)$.

3.1 QAPs

We assume familiarity with quadratic arithmetic programs [GGPR13], but briefly describe the necessary definitions. A QAP $\mathscr{Q}$ of size m, degree n, with ℓ public inputs over $\mathbb{F}$ is defined by a set of univariate polynomials $\left\{\{A_i(X), B_i(X), C_i(X)\}_{i\in[0..m]}, Z(X)\right\}$ where $A_i, B_i, C_i \in \mathbb{F}[X]$ have degree smaller than n, and $Z \in \mathbb{F}[X]$ has degree exactly n. We say $x = (x_1, \ldots, x_\ell) \in \mathbb{F}^\ell, \omega = (x_{\ell+1}, \ldots, x_m) \in \mathbb{F}^{m-\ell}$ *satisfy* $\mathscr{Q}$, if when defining $x_0 = 1$, $A := \sum_{i=0}^m x_i \cdot A_i, B := \sum_{i=0}^m x_i \cdot B_i$, and $C := \sum_{i=0}^m x_i \cdot C_i$; then the polynomial $P := A \cdot B - C$ will be divisble by Z.

3.2 Randomizing QAP witnesses

When describing our SNARK in the next section we will assume the sequences of QAP polynomials were extended in the following way:

$$A_{m+1} = B_{m+2} = Z, A_{m+2} = B_{m+1} = C_{m+1} = C_{m+2} = 0.$$

[2] For simplicity, we present our SNARK with only statistical zero-knowledge, but a slight complication of the construction can give perfect zero-knowledge.

(note in particular that we allow these new polynomials to be of degree n while the former are of degree smaller than n.) For any values $x_{m+1}, x_{m+2} \in \mathbb{F}$, $x, (\omega, x_{m+1}, x_{m+2})$ satisfy the extended QAP if and only if x, ω satisfiy the original one. When describing the prover algorithm in the next section we will assume the values of x_{m+1}, x_{m+2} in the prover's witness ω have been chosen uniformly, and also reindex and denote by m the total number of QAP polynomials after this extension.

4 Description of our SNARK

Let R be the relation of pairs (x, ω) such that x, ω satisfy $\mathscr{Q}$. We proceed to describe our zk-SNARK for the relation R.

Key Generation:

1. Define, for $i \in [0..m]$, the tri-variate polynomial $K_i(\tau, \beta_A, \beta_B) := \beta_B A_i(\tau) + \beta_A B_i(\tau) + C_i(\tau)$. And the rational map

$$\sigma_1(\tau, \beta_A, \beta_B, \delta) := \left\{\tau^i, \tau^i/\delta, \beta_A \tau^i, \beta_B \tau^i\right\}_{i \in [0..n]} \cup \left\{K_i(\tau, \beta_A, \beta_B)/\delta\right\}_{i \in [\ell+1..m]}$$

2. For $\tau, \alpha_A, \alpha_B, \in \mathbb{F}$, define $\sigma_2(\tau, \alpha_A, \alpha_B) := \left\{\alpha_A \tau^i, \alpha_B \tau^i\right\}_{i \in [0..n]}$

3. Choose uniform $\tau, \beta_A, \beta_B, \delta, \alpha_A, \alpha_B \in \mathbb{F}$ and output

$$\sigma := [\sigma_1(\tau, \beta_A, \beta_B, \delta), \sigma_2(\tau, \alpha_A, \alpha_B)]$$

$$r_{\text{ext}} = (\alpha_A, \alpha_B), r_{\text{sim}} = (\tau, \alpha_A, \alpha_B, \beta_A, \beta_B, \delta)$$

Prover:

The prover P has in his hand a QAP solution $(x_0 = 1, x_1, \ldots, x_m)$ that coincides with the public input $x = (x_1, \ldots, x_\ell)$ and satisfies the following: If we define $A := \sum_{i=0}^m x_i \cdot A_i$, $B := \sum_{i=0}^m x_i \cdot B_i$, and $C := \sum_{i=0}^m x_i \cdot C_i$; then the polynomial $P := A \cdot B - C$ will be divisible by the target polynomial Z, and P can compute the polynomial H of degree at most n with $P = H \cdot Z$.

Given the proving key, P computes as linear combinations of the proving key elements $[\pi_A], [\pi_B], [\pi_D], [\pi_K]$ where

1. $\pi_A := A(\tau)$.
2. $\pi_B := B(\tau)$.
3. $\pi_D := \alpha_A A(\tau) + \alpha_B B(\tau)$.
4. $\pi_K := \left(\sum_{i=\ell+1}^m K_i(\tau, \beta_A, \beta_B) + H(\tau)Z(\tau)\right)/\delta$.

and outputs $\pi = ([\pi_A], [\pi_B], [\pi_D], [\pi_K])$.

Verifier:

For $x = (x_1, \ldots, x_\ell)$, denote the "public input component"

$$\mathsf{PI}(x) := K_0(\tau, \beta_A, \beta_B) + \sum_{i=1}^{\ell} x_i K_i(\tau, \beta_A, \beta_B).$$

The verifier, using pairings and the verification key, checks the following.

1. $\pi_D = \alpha_A \cdot \pi_A + \alpha_B \cdot \pi_B$.
2. $(\beta_A + \pi_A) \cdot (\beta_B + \pi_B) = \mathsf{PI}(x) + \pi_K \cdot \delta + \beta_A \cdot \beta_B$.

Simulator $(\phi, x, r_{\mathrm{sim}} = (\tau, \alpha_A, \alpha_B, \beta_A, \beta_B, \delta))$**:**

1. Choose $\pi_A, \pi_B \in \mathbb{F}$ uniformly.
2. Let $\pi_D := \alpha_A \pi_A + \alpha_B \pi_B$.
3. Let $\pi_K := \frac{1}{\delta}((\beta_A + \pi_A)(\beta_B + \pi_B) - \mathsf{PI}(x))$. Output $([\pi_A], [\pi_B], [\pi_D], [\pi_K])$.

In the next section we prove the following:

Theorem 4.1. *Under the n-PKE and $(4, 2n)$-SPAN assumptions the above scheme is a zk-SNARK for the relation R.*

5 Security proof

The completeness and zero-knowledge properties are easily verifiable from the construction. We thus focus on knowledge soundness.

Description of extractor Fix an efficient $\mathcal{A}$. We need to describe the corresponding extractor E. Suppose that $\mathcal{A}$ has outputted (x, π) such that $\mathsf{V}(\sigma, x, \pi) = \mathsf{acc}$. Let $\pi = ([\pi_A], [\pi_B], [\pi_D], [\pi_K])$.

In particular, we have

$$\alpha_A \cdot \pi_A + \alpha_B \cdot \pi_B = \pi_D.$$

This implies by Lemma 2.3 that given $r_{\mathrm{ext}} = \alpha_B$ and $\mathsf{rand}_{\mathcal{A}}$, E can extract e.w.p. $\mathsf{negl}(\lambda)$ a polynomial $A \in \mathbb{F}[X]$ of degree at most n such that

$$\pi_A = A(\tau).$$

E uses linear algebra to determine if the polynomial $A_{\mathrm{mid}}(X) := A(X) - \sum_{i=0}^{\ell} x_i A_i(X)$ is in the span of the QAP polynomials $\{A_i\}_{i \in [\ell+1..m]}$. If not, E aborts. Otherwise, let $x_{\ell+1}, \ldots, x_m \in \mathbb{F}$ be such that $A_{\mathrm{mid}}(X) = \sum_{i=\ell+1}^{m} x_i A_i(X)$, and let $\omega := (x_{\ell+1}, \ldots, x_m)$. E checks if $(x, \omega) \in R$ and if so outputs ω as a witness for x.

Analyzing E's failure probability Let η be the probability that $\mathcal{A}$ outputs a verifying (x,π) but E doesn't output a valid witness ω for x. To prove knowledge soundness we must show that $\eta = \mathsf{negl}(\lambda)$. We will construct an efficient $\mathcal{A}^*$ that solves $(4,2n)$-SPAN with probability $\eta - \mathsf{negl}(\lambda)$ thus implying $\eta = \mathsf{negl}(\lambda)$ under the $(4,2n)$-SPAN assumption.

The challenge $\mathcal{A}^*$ will receive will be $T_1 := [\sigma_1(\tau,\beta_A,\beta_B,\delta)]$ for uniform $\tau,\beta_A,\beta_B,\delta \in \mathbb{F}$.

Recall that

$$\sigma_1(\tau,\beta_A,\beta_B,\delta) = \left\{\tau^i, \tau^i/\delta, \beta_A\tau^i, \beta_B\tau^i\right\}_{i\in[0..n]} \cup \left\{K_i(\tau,\beta_A,\beta_B)/\delta\right\}_{i\in[0..m]}$$

$\mathcal{A}^*$ samples uniform $\alpha_A,\alpha_B \in \mathbb{F}$ and computes $T_2 := [\sigma_2(\tau,\alpha_A,\alpha_B)]$. Define $\sigma := (T_1,T_2)$. $\mathcal{A}^*$ runs $\mathcal{A}(\mathsf{rand}_{\mathcal{A}},\sigma)$ to get output (x,π) and checks whether $\mathsf{V}(\sigma,x,\pi) = \mathsf{acc}$. If the check fails $\mathcal{A}^*$ aborts. Otherwise, denote $\pi = ([\pi_A],[\pi_B],[\pi_D],[\pi_K])$. We again have

$$\alpha_A \cdot \pi_A + \alpha_B \cdot \pi_B = \pi_D$$

This implies by Lemma 2.3 that given α_A,α_B and $\mathsf{rand}_{\mathcal{A}}$, $\mathcal{A}^*$ can extract e.w.p. $\mathsf{negl}(\lambda)$ polynomials $A,B \in \mathbb{F}[X]$ of degree at most n such that

$$\pi_A = A(\tau), \pi_B = B(\tau).$$

If one of the extractions failed, $\mathcal{A}^*$ aborts.

$\mathcal{A}^*$ now computes the 4-variate rational function

$$C(X,X_A,X_B,X_\delta) := (A(X)B(X) + X_AB(X) + X_BA(X) - \mathsf{PI}(X,X_A,X_B))/X_\delta,$$

where $\mathsf{PI}(X,X_A,X_B) := \sum_{i=0}^{\ell} x_i \cdot K_i(X,X_A,X_B)$. Rearranging the second verification equation we see that we have

$$\pi_K = (\pi_A\pi_B + \beta_A\pi_B + \beta_B\pi_A - \mathsf{PI}(X))/\delta = C(\tau,\beta_A,\beta_B,\delta).$$

Denote $U := \mathsf{span}(\sigma_1(X,X_A,X_B,X_\delta)) \subseteq \mathbb{F}(X,X_A,X_B,X_\delta)$. If $C \notin U$, $\mathcal{A}^*$ outputs $(C,[\pi_K])$ as a response to the $(4,2n)$-SPAN challenge.

We show that when π is a valid proof but $C \in U$, then E outputs a valid witness:

$C \in U$ implies there are polynomials $f, f_A, f_B, H \in \mathbb{F}[X]$ of degree at most n, together with coefficients $x_{\ell+1},\ldots,x_m \in \mathbb{F}$, such that

$$\begin{aligned}&(A(X)B(X) + X_AB(X) + X_BA(X) - \mathsf{PI}(X,X_A,X_B))/X_\delta\\ &= f(X) + f_A(X) + f_B(X) + H(X)Z(X)/X_\delta + \left(\sum_{i=\ell+1}^{m} x_i \cdot K_i(X,X_A,X_B)\right)/X_\delta.\end{aligned}$$

We multiply by X_δ and add $\mathsf{PI}(X,X_A,X_B)$ to get

$$\begin{aligned}&A(X)B(X) + X_AB(X) + X_BA(X)\\ &= (f(X) + f_A(X) + f_B(X))X_\delta + H(X)Z(X) + \sum_{i=0}^{m} x_i \cdot K_i(X,X_A,X_B).\end{aligned}$$

Let us think of the above as a polynomial identity in the variables X_A, X_B, X_δ with coefficients in $\mathbb{F}[X]$. Since the constant coefficient, that of X_A and that of X_B must be identical on both sides, we have

1. $A(X)B(X) = \sum_{i=0}^{m} x_i \cdot C_i(X) + H(X)Z(X)$.
2. $B(X) = \sum_{i=0}^{m} x_i \cdot B_i(X)$.
3. $A(X) = \sum_{i=0}^{m} x_i \cdot A_i(X)$.

This exactly means that $\omega := (x_{\ell+1}, \ldots, x_m)$ is a valid witness for x. And ω exactly corresponds to the output of E.

Acknowledgements

We thank Jens Groth and Mary Maller for helpful discussions on the subject of this paper.

References

[BCPR13] N. Bitansky, R. Canetti, O. Paneth, and A. Rosen. Indistinguishability obfuscation vs. auxiliary-input extractable functions: One must fall. *IACR Cryptology ePrint Archive*, 2013:641, 2013.

[DFGK14] G. Danezis, C. Fournet, J. Groth, and M. Kohlweiss. Square span programs with applications to succinct NIZK arguments. In *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, pages 532–550, 2014.

[GGPR13] R. Gennaro, C. Gentry, B. Parno, and M. Raykova. Quadratic span programs and succinct nizks without pcps. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 626–645, 2013.

[GM17] J. Groth and M. Maller. Snarky signatures: Minimal signatures of knowledge from simulation-extractable snarks. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 581–612, 2017.

[Gro10] J. Groth. Short pairing-based non-interactive zero-knowledge arguments. In *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, pages 321–340, 2010.

[Gro16] J. Groth. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, pages 305–326, 2016.

[GW11] C. Gentry and D. Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 99–108, 2011.

[Nao03] M. Naor. On cryptographic assumptions and challenges. In *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, pages 96–109, 2003.

[PHGR16] B. Parno, J. Howell, C. Gentry, and M. Raykova. Pinocchio: nearly practical verifiable computation. *Commun. ACM*, 59(2):103–112, 2016.