A Compact Split Non-interactive Linear Proof for Boolean Circuit Satisfiability

Jens Groth

August 26, 2026

Abstract

We construct a pairing-based zk-SNARK with one element from each source group in the generic Type-III bilinear-group model. Our construction does not use cryptographic hash functions, and the analysis does not rely on the random oracle model. The core of our construction is a non-interactive linear proof (NILP) for a sparse homogeneous linear system over signed Boolean values. The NILP is split, i.e., the field elements in the common reference string and the proof are partitioned into two separate tracks, which facilitates compilation into a Type-III pairing argument. Many algebraic pairing-based SNARKs can be viewed as compilations of split NILPs. This is the first split NILP with a single field element response on each side for a relation family whose induced language is NP-complete, and the first publicly verifiable preprocessing SNARK in the plain generic Type-III bilinear-group model whose proof contains one element from each source group. Verification is efficient; it requires work proportional to the public input. Proving is less efficient and uses a quadratic-size common reference string.

1 Introduction

Pairing-based succinct non-interactive arguments of knowledge (SNARKs) achieve small proofs by evaluating polynomial identities in the exponent and checking quadratic relations between the proof elements with bilinear maps. Groth's three-element construction arranges the proof as two elements in the first source group and one in the second. For disclosure-free generic Type-III pairing-based arguments for hard decisional relation families, his lower bound requires proof elements from both source groups [Gro16]. This leaves a natural question: can such an argument have exactly one proof element in each source group?

Many pairing-based SNARKs can be viewed as instantiations of an information-theoretic abstraction called a non-interactive linear proof (NILP) defined over a finite field. Its common reference string generator samples field elements; without seeing the sampled CRS, the proof-matrix algorithm picks coefficients for linear combinations; the prover then applies those coefficients to the CRS; and the verifier checks the resulting proof using quadratic equations over the field. In a split NILP, the common reference string and the proof each have a left and a right part, each proof part is a linear function only of its corresponding reference-string half, and the verifier performs cross-bilinear tests. A (disclosure-free) split NILP can be compiled into a Type-III pairing argument in the generic bilinear-group model [Gro16]. The central problem we address is to construct a split NILP with a single field element response on each side for a relation family whose induced language is NP-complete.

Our construction has three algebraic layers. First, Boolean circuit satisfiability is reduced to a sparse homogeneous linear system over signed Boolean coordinates. One residual coordinate per gate turns each NAND-gate constraint into a linear equation, leaving Booleanity as the only

nonlinear condition on the coordinates. Second, we introduce factorization span programs (FSPs). An FSP specifies two fixed linear spaces of polynomials and an input-linear family of public targets. Every target polynomial, defined by the public input, lies in the span of pairwise products from the two spaces, but acceptance requires the stronger condition that the target polynomial admit a single exact factorization into two factors, one from each space. Third, random evaluation of an FSP gives a split NILP with one field response on each side.

The main technical contribution is the FSP. The intuition behind it is to test Booleanity through unique polynomial factorization. To give an example, consider the polynomial $H(X) = X^2 - 1$ over a finite field $\mathbb{F}$. If $A(X)B(X) = H(X)$ with $A, B \in \mathbb{F}^{\leq 1}[X]$, then $A(X)$ must be of the form $v(X - b)$ for some $v \in \mathbb{F}^\times$ and $b \in \{-1, +1\}$. Using the convention $-1 = \mathsf{false}$ and $+1 = \mathsf{true}$, this means that $A(X) = v(X - b)$ represents a unique Boolean value b.

Generalizing this idea, we can represent many Boolean values $b_1, \ldots, b_n$ by a product of linear factors

$$A(X, Z) = \prod_{i=1}^{n} \big((Z - \omega_i)X - b_i\big), \qquad b_i \in \{-1, +1\},$$

where $\omega_1, \ldots, \omega_n$ are distinct points in the field used as tags. Picking the complementary polynomial

$$B(X, Z) = \prod_{i=1}^{n} \big((Z - \omega_i)X + b_i\big),$$

we have $A(X, Z)B(X, Z) = H(X, Z)$ for the fixed polynomial

$$H(X, Z) = \prod_{i=1}^{n} \big((Z - \omega_i)^2 X^2 - 1\big).$$

Unique factorization does not quite suffice to guarantee that $A(X, Z)$ is well formed. Even if $A(X, Z)$ has degree n in X, an adversary might, for instance, include both $(Z - \omega_1)X - 1$ and $(Z - \omega_1)X + 1$ as factors, whereas we want to force the adversary to pick exactly one factor for each tag. We therefore require A to belong to a specific linear space of polynomials $\mathcal{U}$ whose leading-coefficient restriction ensures a unique choice of Boolean values. The restrictions defining $\mathcal{U}$ can also impose linear constraints on $b_1, \ldots, b_n$. Finally, the full construction introduces a third indeterminate Y, and the target polynomial depends on the instance $\mathbf{u}$; we write it as $H_{\mathbf{u}}(X, Y, Z)$.

The split NILP over a finite field has perfect completeness, statistical knowledge soundness against split-affine strategies, and disclosure-freeness from the Schwartz–Zippel lemma. Multiplicative rerandomization gives perfect simulation under a statistically close simulated CRS, and hence statistical composable zero knowledge.

Using the split NILP as the information-theoretic core, we can build a pairing-based preprocessing zk-SNARK in the generic Type-III bilinear-group model. The proof has one group element in each source group. While the proof has one fewer group element than Groth's construction, the CRS is quadratic in the circuit size, so the construction is primarily of theoretical interest. The main contribution is that this is the first split NILP with split dimensions $(1, 1)$ for a relation family whose induced language is NP-complete, thus resolving a decade-old question [Gro16, Section 1].

2 Boolean circuit satisfiability and homogeneous linear equations

Consider a Boolean circuit $\mathcal{C}$, represented as a directed acyclic graph with G fan-in-two NAND gates and a single output. Topologically order the gates so that gate G is the output gate, fix its output to true, and eliminate that output-wire variable. Let W denote the number of remaining

input and internal wires. Circuit satisfiability asks whether there exists a Boolean assignment to the input wires such that the circuit outputs true. This is known to be an NP-complete problem.

For a fixed circuit $\mathcal{C}$, we can designate ℓ of the input wires as public. We can then ask whether, for a given instance consisting of truth-value assignments to the public inputs, there exists a Boolean assignment to the remaining $W - \ell$ witness wires such that the circuit outputs true. This defines a finite relation $\mathcal{R}_\mathcal{C}$ between instances and witnesses, and a language $\mathcal{L}_\mathcal{C}$ of instances admitting a witness. NP-completeness applies to the resulting parameterized family: a polynomial-time constructible family of universal circuits can take the description of any suitably smaller circuit as part of its public input and evaluate that circuit on the remaining inputs [Val76].

We can reduce a Boolean circuit $\mathcal{C}$ to a sparse homogeneous linear system over signed Boolean values in a finite field $\mathbb{F}$. The reduction introduces one additional residual coordinate per gate, so we set

$$N = W + G, \qquad n = N - \ell.$$

Here N is the number of nonconstant coordinates in the linear system, ℓ is the number of public input coordinates matching the circuit instance, and n is the number of remaining coordinates. We will work over a finite field $\mathbb{F}$ of size

$$q = |\mathbb{F}|,$$

and require $\mathrm{char}(\mathbb{F}) > 3$ and $q > N$. Boolean values are represented by signed coordinates in $\mathbb{F}$:

$$-1 = \mathsf{false}, \qquad +1 = \mathsf{true}.$$

Let $a_1, \ldots, a_W \in \{-1, +1\}$ represent truth values of the circuit wires. After relabeling wires if necessary, the public input coordinates are

$$\mathbf{u} = (a_1, \ldots, a_\ell).$$

Following [GOS06], we can arithmetize the gates in the circuit. For each NAND gate g, with input coordinates a_{i_g}, a_{j_g} and output coordinate a_{k_g}, introduce a fresh signed residual coordinate a_{W+g} and impose

$$-1 + a_{i_g} + a_{j_g} + 2a_{k_g} - a_{W+g} = 0.$$

For each gate, if and only if the gate is evaluated correctly, there exists a unique residual value $a_{W+g} \in \{-1, +1\}$ satisfying this equation. Specifically, the affine residual

$$r_g = -1 + a_{i_g} + a_{j_g} + 2a_{k_g}$$

belongs to $\{-1, +1\}$ for a correct NAND evaluation and to $\{-5, -3, 3\}$ for an incorrect evaluation. Setting $a_{W+g} = r_g$ therefore gives the unique signed residual satisfying the gate equation exactly when the gate is evaluated correctly.

It will be helpful to have a constant coordinate $a_0 = 1$ to homogenize the linear system. We represent the eliminated accepting output by a_0; equivalently, set $k_G = 0$. For every other gate, k_g is an ordinary nonconstant wire index. Thus each NAND gate becomes one homogeneous linear equation $-a_0 + a_{i_g} + a_{j_g} + 2a_{k_g} - a_{W+g} = 0$ with at most five nonzero coefficients. The word *homogeneous* refers to the augmented variables: the apparent constant term is the coefficient of a_0, whose value is subsequently fixed to 1.

We have focused on NAND gates since NAND is universal, but similar affine-map characterizations exist for all fan-in-two Boolean gates [DFGK14]. Homogenizing each affine expression with a_0 and introducing a signed residual coordinate gives the same kind of homogeneous gate equation, again with at most five nonzero coefficients.

Each gate equation can be written as a column of a matrix $D \in \mathbb{F}^{(N+1)\times G}$, where the rows correspond to the augmented coordinates $(a_0, a_1, \ldots, a_W, a_{W+1}, \ldots, a_N)$. For the circuit reduction, the columns are automatically linearly independent: gate column g has a nonzero entry in its dedicated residual row $W+g$, where every other gate column is zero in this row. Hence $\mathrm{rank}(D) = G$ for the concrete circuit matrix. The Boolean circuit satisfiability problem can therefore be reduced to finding a signed solution to a homogeneous linear system $(a_0, \ldots, a_N)D = 0$ with $a_0 = 1$ and $a_1, \ldots, a_N \in \{-1, +1\}$, subject to the requirement that the first ℓ coordinates are fixed to the instance $\mathbf{u}$. The reduction is efficient and the resulting linear system has at most $5N$ nonzero entries, so D is sparse.

Write

$$\mathbf{u} = (a_1, \ldots, a_\ell), \qquad \mathbf{w} = (a_{\ell+1}, \ldots, a_N), \qquad \overline{\mathbf{a}}(\mathbf{u}, \mathbf{w}) = (1, \mathbf{u}, \mathbf{w}).$$

The Boolean homogeneous linear relation defined by $\mathbb{F}, \ell, G, W$, and $D \in \mathbb{F}^{(N+1)\times G}$, with $N = G+W$ and $n = N - \ell$, is

$$\mathcal{R} = \left\{ (\mathbf{u}, \mathbf{w}) \in \{-1, +1\}^\ell \times \{-1, +1\}^n : \overline{\mathbf{a}}(\mathbf{u}, \mathbf{w})D = 0 \right\}.$$

It induces the language

$$\mathcal{L} = \left\{ \mathbf{u} \in \{-1, +1\}^\ell : \exists\, \mathbf{w} \in \{-1, +1\}^n \text{ such that } (\mathbf{u}, \mathbf{w}) \in \mathcal{R} \right\}.$$

Thus $\mathbf{u}$ is the instance, the statement is the proposition $\mathbf{u} \in \mathcal{L}$, and $\mathbf{w}$ is a witness establishing that statement. In the following, we will construct a proof system for this relation.

3 Split non-interactive linear proofs

Following Groth [Gro16], we recall the NILP syntax in the row-vector convention used throughout this paper. A split NILP for a relation $\mathcal{R}$ exposes the following algorithms.

Definition 3.1 (Split NILP syntax)**.** Setup produces two public CRS row vectors and an associated trapdoor,

$$((\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2), \mathsf{td}) \leftarrow \mathsf{Setup}(\mathcal{R}), \qquad \boldsymbol{\sigma}_1 \in \mathbb{F}^{d_1}, \quad \boldsymbol{\sigma}_2 \in \mathbb{F}^{d_2}.$$

On an instance and witness $(\mathbf{u}, \mathbf{w}) \in \mathcal{R}$, the randomized proof-matrix algorithm computes

$$(P_1, P_2) \leftarrow \mathsf{ProofMatrix}(\mathcal{R}, \mathbf{u}, \mathbf{w}), \qquad P_1 \in \mathbb{F}^{k_1 \times d_1}, \quad P_2 \in \mathbb{F}^{k_2 \times d_2}.$$

Crucially, ProofMatrix does not receive the sampled CRS. The prover algorithm does receive it and applies each proof matrix only to its corresponding CRS half:

$$(\boldsymbol{\pi}_1, \boldsymbol{\pi}_2) \leftarrow \mathsf{Prove}(\mathcal{R}, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \mathbf{u}, \mathbf{w}), \qquad \boldsymbol{\pi}_1 = \boldsymbol{\sigma}_1 P_1^T, \qquad \boldsymbol{\pi}_2 = \boldsymbol{\sigma}_2 P_2^T.$$

The deterministic test algorithm computes

$$(T_1, \ldots, T_\eta) \leftarrow \mathsf{Test}(\mathcal{R}, \mathbf{u}), \qquad T_h \in \mathbb{F}^{(d_1+k_1)\times(d_2+k_2)},$$

and Vfy first rejects unless $\mathbf{u}$ belongs to the public input domain $\mathcal{X}$ of $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$, and otherwise accepts precisely when every cross-bilinear test vanishes:

$$(\boldsymbol{\sigma}_1, \boldsymbol{\pi}_1) T_h (\boldsymbol{\sigma}_2, \boldsymbol{\pi}_2)^T = 0 \qquad (1 \le h \le \eta).$$

The associated completeness, knowledge soundness, disclosure freeness, and zero-knowledge requirements are defined next.

We use the following fixed-parameter security notions. They quantify exact error probabilities for the fixed relation and field under consideration and therefore require neither a security parameter nor an asymptotic notion of negligibility. For zero knowledge, we separate closeness of the real and simulated CRS distributions from perfect proof simulation under the simulated CRS, following Groth's formulation of composable zero knowledge [Gro06].

Definition 3.2 (Fixed-parameter security)**.** A split NILP has the following properties.

1. It has *Perfect completeness* if, for every $(\mathbf{u}, \mathbf{w}) \in \mathcal{R}$, an honestly generated proof is accepted with probability one over setup and prover randomness.

2. It has *split-affine statistical knowledge error at most* ε if there is an efficient extractor Ext such that the following holds. Let any possibly randomized strategy, without seeing the sampled CRS, choose an instance $\mathbf{u}$ and proof matrices P_1, P_2 of the prescribed dimensions. For an independently generated setup, set $\boldsymbol{\pi}_i = \boldsymbol{\sigma}_i P_i^T$. Then

$$\Pr\big[\mathsf{Vfy}(\mathcal{R}, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \mathbf{u}, (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2)) = 1 \ \wedge \ (\mathbf{u}, \mathsf{Ext}(\mathcal{R}, \mathbf{u}, P_1, P_2)) \notin \mathcal{R}\big] \le \varepsilon.$$

   The word *affine* allows constant terms; when each CRS half contains the coordinate 1, they are represented by ordinary linear proof matrices.

3. It has *disclosure disagreement at most* δ if, for every setup-independent matrix $T \in \mathbb{F}^{d_1 \times d_2}$ and two independent setups $(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2)$ and $(\boldsymbol{\sigma}_1', \boldsymbol{\sigma}_2')$,

$$\Pr\Big[\boldsymbol{\sigma}_1 T \boldsymbol{\sigma}_2^T = 0 \text{ and } \boldsymbol{\sigma}_1' T (\boldsymbol{\sigma}_2')^T \neq 0\Big] \le \delta.$$

   We say the system has a δ-*disclosure-free* CRS.

4. It has *statistical composable zero knowledge with error at most* ζ if there are efficient algorithms $\mathsf{SimSetup}$ and Sim satisfying the following two properties.

   (a) The real and simulated public CRS distributions are statistically close. Namely, if

$$((\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2), \mathsf{td}) \leftarrow \mathsf{Setup}(\mathcal{R}) \quad \text{and} \quad ((\widehat{\boldsymbol{\sigma}}_1, \widehat{\boldsymbol{\sigma}}_2), \widehat{\mathsf{td}}) \leftarrow \mathsf{SimSetup}(\mathcal{R}),$$

   then

$$\Delta((\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2), (\widehat{\boldsymbol{\sigma}}_1, \widehat{\boldsymbol{\sigma}}_2)) \le \zeta.$$

   Only the public CRS distributions are compared; the setup trapdoors need not have the same distribution. In fact, in real usage td is not revealed and just represents the internal variables used by the setup algorithm.

   (b) Proof simulation is perfect under the simulated CRS, even given the simulation trapdoor. Give $(\widehat{\boldsymbol{\sigma}}_1, \widehat{\boldsymbol{\sigma}}_2, \widehat{\mathsf{td}})$ to any adaptive environment $\mathcal{A}$, which chooses an instance–witness pair $(\mathbf{u}, \mathbf{w})$ as a function of the simulated CRS and trapdoor. If the pair is invalid, both experiments return $\perp$. Otherwise, one experiment returns a fresh honest proof

$$\mathsf{Prove}(\mathcal{R}, \widehat{\boldsymbol{\sigma}}_1, \widehat{\boldsymbol{\sigma}}_2, \mathbf{u}, \mathbf{w}),$$

   while the other returns

$$\mathsf{Sim}(\mathcal{R}, \widehat{\boldsymbol{\sigma}}_1, \widehat{\boldsymbol{\sigma}}_2, \widehat{\mathsf{td}}, \mathbf{u}),$$

   without giving $\mathbf{w}$ to Sim. The two full views, including the environment's randomness, the simulated CRS and trapdoor, its chosen pair, the reply, and its final output, must be identically distributed.

Replacing the real CRS by the simulated CRS and then using perfect proof simulation shows that this definition implies the usual real-versus- simulated view distance of at most ζ. This is the statistical version of Groth's *composable zero knowledge*, which is stronger than standard zero knowledge and implies unbounded adaptive zero knowledge under reuse of the same CRS [Gro06]. The property is *perfect composable zero knowledge* when $\zeta = 0$.

The pair (k_1, k_2) is the split proof dimension. Groth showed that a disclosure-free generic Type-III pairing argument for a relation generator with hard decisional problems must contain proof elements from both source groups. His construction has split proof dimension $(2, 1)$. We construct a $(1, 1)$-split NILP for a relation family whose induced language is NP-complete, attaining the smallest split dimensions consistent with this lower bound.

4 Factorization span programs

This section isolates the polynomial object underlying the construction. It plays the same organizational role that other polynomial objects such as QSP, QAP, SSP and SAP play in other pairing-based proof systems: it is an algebraic relation, not yet a proof system. A generic random-evaluation compiler will later turn it into a split NILP.

Unless stated otherwise, vectors are row vectors. For an integer $d \geq 0$, write

$$\mathbb{F}^{\leq d}[Z] = \{f(Z) \in \mathbb{F}[Z] : \deg f \leq d\}.$$

For a multivariate polynomial, $\deg_{\mathrm{tot}}$ denotes total degree.

4.1 Definition

Let

$$\mathcal{P} = \mathbb{F}[X_1, \ldots, X_s]$$

be a polynomial ring. Let $\mathcal{U}, \mathcal{V} \subseteq \mathcal{P}$ be finite-dimensional $\mathbb{F}$-linear spaces with fixed row-vector bases

$$\mathbf{U} = (U_1, \ldots, U_{d_U}), \qquad \mathbf{V} = (V_1, \ldots, V_{d_V}).$$

We require $1 \in \mathcal{U} \cap \mathcal{V}$ and choose $U_1 = V_1 = 1$.

Definition 4.1 (Factorization span program)**.** An ℓ-input *factorization span program* (FSP) consists of the spaces and bases above, target polynomials

$$H_0, H_1, \ldots, H_\ell \in \mathcal{P},$$

and matrices

$$M_0, M_1, \ldots, M_\ell \in \mathbb{F}^{d_U \times d_V}$$

such that

$$\mathbf{U}(\mathbf{X}) M_i \mathbf{V}(\mathbf{X})^T = H_i(\mathbf{X}) \qquad (0 \leq i \leq \ell). \tag{1}$$

For a public input $\mathbf{u} = (u_1, \ldots, u_\ell) \in \mathbb{F}^\ell$, define

$$H_{\mathbf{u}}(\mathbf{X}) = H_0(\mathbf{X}) + \sum_{i=1}^{\ell} u_i H_i(\mathbf{X}), \qquad M_{\mathbf{u}} = M_0 + \sum_{i=1}^{\ell} u_i M_i. \tag{2}$$

The program accepts $\mathbf{u}$ if there exist $\mathsf{A} \in \mathcal{U}$ and $\mathsf{B} \in \mathcal{V}$ satisfying the formal identity

$$\mathsf{A}(\mathbf{X})\mathsf{B}(\mathbf{X}) = H_{\mathbf{u}}(\mathbf{X}). \tag{3}$$

The affine-linear dependence of $H_{\mathbf{u}}$ and $M_{\mathbf{u}}$ on the public input is what we refer to as *input linearity*.

Remark 4.2 (Rank-one preimages)**.** Define the multiplication map

$$\mu : \mathbb{F}^{d_U \times d_V} \longrightarrow \mathcal{P}, \qquad \mu(M) = \mathbf{U} M \mathbf{V}^T.$$

The matrix $M_{\mathbf{u}}$ is an arbitrary preimage of the public target. Writing

$$\mathsf{A} = \mathbf{c}_A \mathbf{U}^T, \qquad \mathsf{B} = \mathbf{c}_B \mathbf{V}^T,$$

an accepting factorization gives the rank-one preimage $\mathbf{c}_A^T \mathbf{c}_B \in \mathbb{F}^{d_U \times d_V}$. Thus an FSP asks whether a target known to lie in a product span $\mathcal{U}\mathcal{V}$ has a rank-one preimage under μ.

4.2 Realizing an external relation

The native witness of an FSP is a pair of polynomials. To use the object for an NP relation, we require efficient translations between ordinary witnesses and polynomial factors.

Definition 4.3 (FSP realization)**.** Let $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{W}$, where $\mathcal{X} \subseteq \mathbb{F}^\ell$. An FSP *realizes* $\mathcal{R}$ if there are polynomial-time algorithms Enc and Dec such that:

(i) if $(\mathbf{u}, \mathbf{w}) \in \mathcal{R}$, then

$$(\mathsf{A}, \mathsf{B}) \leftarrow \mathsf{Enc}(\mathbf{u}, \mathbf{w})$$

satisfies $\mathsf{A} \in \mathcal{U}$, $\mathsf{B} \in \mathcal{V}$, and $\mathsf{AB} = H_{\mathbf{u}}$;

(ii) if $\mathbf{u} \in \mathcal{X}$ and $\mathsf{A} \in \mathcal{U}$, $\mathsf{B} \in \mathcal{V}$ satisfy $\mathsf{AB} = H_{\mathbf{u}}$, then

$$\mathbf{w} \leftarrow \mathsf{Dec}(\mathbf{u}, \mathsf{A}, \mathsf{B})$$

satisfies $(\mathbf{u}, \mathbf{w}) \in \mathcal{R}$.

For an FSP, define the degree parameter

$$d_{\mathsf{FSP}} = \max \left\{ \max_{i,j} \deg_{\mathrm{tot}}(U_i V_j), \max_{0 \le k \le \ell} \deg_{\mathrm{tot}} H_k \right\}. \tag{4}$$

4.3 Generic compiler to a split NILP

The FSP abstraction makes the information-theoretic compiler immediate.

Theorem 4.4 (FSP compiler)**.** *Suppose an ℓ-input FSP over $\mathbb{F}$ realizes a relation $\mathcal{R}$ and has degree parameter d_{FSP}. Random evaluation gives a $(1,1)$-split NILP with perfect completeness, split-affine statistical knowledge error at most d_{FSP}/q, and disclosure disagreement at most d_{FSP}/q.*

Proof. Sample independently and uniformly

$$\mathbf{x} = (x_1, \ldots, x_s) \leftarrow \mathbb{F}^s$$

and let the two CRS halves be

$$\boldsymbol{\sigma}_1 = \mathbf{U}(\mathbf{x}) \in \mathbb{F}^{d_U}, \qquad \boldsymbol{\sigma}_2 = \mathbf{V}(\mathbf{x}) \in \mathbb{F}^{d_V}. \tag{5}$$

Given $(\mathbf{u}, \mathbf{w}) \in \mathcal{R}$, compute $(\mathsf{A}, \mathsf{B}) \leftarrow \mathsf{Enc}(\mathbf{u}, \mathbf{w})$, express the two polynomials in the fixed bases as $\mathbf{c}_A \in \mathbb{F}^{d_U}$ and $\mathbf{c}_B \in \mathbb{F}^{d_V}$, and compute two proof halves

$$A = \mathsf{A}(\mathbf{x}) = \mathbf{c}_A \boldsymbol{\sigma}_1^T, \qquad B = \mathsf{B}(\mathbf{x}) = \mathbf{c}_B \boldsymbol{\sigma}_2^T. \tag{6}$$

The verifier performs any explicit domain check required for $\mathbf{u} \in \mathcal{X}$ and accepts if

$$AB = \boldsymbol{\sigma}_1 M_{\mathbf{u}} \boldsymbol{\sigma}_2^T. \tag{7}$$

By definition of $M_{\mathbf{u}}$, the right-hand side equals $H_{\mathbf{u}}(\mathbf{x})$, so formal factorization gives perfect completeness.

A split-affine strategy is described by one coefficient row for each CRS half. Since $1 \in \mathcal{U} \cap \mathcal{V}$, these rows define polynomials $\mathsf{A} \in \mathcal{U}$ and $\mathsf{B} \in \mathcal{V}$, including arbitrary affine constants. If

$$\mathsf{A}(\mathbf{X})\mathsf{B}(\mathbf{X}) - H_{\mathbf{u}}(\mathbf{X})$$

is the zero polynomial, Dec returns a witness. Otherwise it is a nonzero polynomial of degree at most d_{FSP} and vanishes at the sampled point with probability at most d_{FSP}/q by Schwartz–Zippel.

Finally, a fixed cross-bilinear zero test on the CRS has formal polynomial

$$P_T(\mathbf{X}) = \mathbf{U}(\mathbf{X}) T \mathbf{V}(\mathbf{X})^T$$

for a setup-independent matrix $T \in \mathbb{F}^{d_U \times d_V}$. It is either formally zero, in which case the disclosure event is impossible, or it vanishes accidentally with probability at most d_{FSP}/q. The event in Definition 3.2 requires the polynomial to vanish at the first sampled point, so its probability is at most d_{FSP}/q. □

5 A factorization span program for signed linear systems

We now construct an FSP realizing the relation from Section 2. The formal variables are X, Y, Z, and their independently sampled setup evaluations will be denoted by x, y, z.

5.1 Public extendability

The target-span argument needs a field-valued extension for every public input, including false Boolean instances.

Definition 5.1 (Public extendability)**.** A homogeneous signed linear system with public–witness partition (ℓ, n) is *publicly extendable* if for every $\mathbf{u} \in \mathbb{F}^\ell$ there is $\mathbf{w}^\star \in \mathbb{F}^n$ such that

$$\overline{\mathbf{a}}(\mathbf{u}, \mathbf{w}^\star) D = 0.$$

No Booleanity is required of $\mathbf{w}^\star$.

Write the rows of D according to the constant, public, and witness coordinates as

$$D = \begin{pmatrix} \mathbf{d}_0 \\ D_{\mathrm{pub}} \\ D_{\mathrm{wit}} \end{pmatrix}, \qquad D_{\mathrm{pub}} = \begin{pmatrix} \mathbf{d}_1 \\ \vdots \\ \mathbf{d}_\ell \end{pmatrix}.$$

Lemma 5.2 (Linear public extension)**.** *If D is publicly extendable, then Gaussian elimination can precompute $\mathbf{b}_0, \dots, \mathbf{b}_\ell \in \mathbb{F}^n$ satisfying*

$$\mathbf{b}_0 D_{\mathrm{wit}} = -\mathbf{d}_0, \qquad \mathbf{b}_i D_{\mathrm{wit}} = -\mathbf{d}_i \quad (1 \le i \le \ell).$$

Consequently,

$$\mathbf{w}^\star(\mathbf{u}) = \mathbf{b}_0 + \sum_{i=1}^{\ell} u_i \mathbf{b}_i \tag{8}$$

is an efficiently computable input-linear extension satisfying $\overline{\mathbf{a}}(\mathbf{u}, \mathbf{w}^\star(\mathbf{u}))D = 0$ for every $\mathbf{u} \in \mathbb{F}^\ell$.

Proof. Public extendability at $\mathbf{u} = \mathbf{0}$ gives a vector $\mathbf{b}_0$ with $\mathbf{b}_0 D_{\mathrm{wit}} = -\mathbf{d}_0$. For each standard basis vector $\mathbf{e}_i \in \mathbb{F}^\ell$, let $\mathbf{c}_i$ be an extension. Then

$$\mathbf{c}_i D_{\mathrm{wit}} = -\mathbf{d}_0 - \mathbf{d}_i,$$

so $\mathbf{b}_i = \mathbf{c}_i - \mathbf{b}_0$ has the required property. These vectors can be found by Gaussian elimination on the fixed matrix D_{wit}. Substitution now proves the final identity. □

Lemma 5.3 (The circuit system is publicly extendable)**.** *The matrix D constructed during the Boolean circuit reduction in Section 2 is publicly extendable.*

Proof. Fix the public coordinates and assign arbitrary field values to the remaining ordinary wires. The column belonging to gate g has coefficient -1 in its dedicated residual row $W + g$, and no other gate column uses that row. The residual coordinates can therefore be solved independently, one gate equation at a time. This is the precise reason a field-valued witness completion always exists. □

The target-span proof below is most convenient for $n \ge 2$. If necessary, append unused witness signs whose rows in D are zero; this changes the witness representation but not the language.

5.2 Tagged factors for witness signs

Choose distinct tag points

$$\omega_1, \dots, \omega_N \in \mathbb{F}.$$

The first ℓ tags are reserved for public-input interpolation. For the witness tags define

$$t(Z) = \prod_{j=\ell+1}^{N} (Z - \omega_j), \qquad t_j(Z) = \frac{t(Z)}{Z - \omega_j} \quad (\ell < j \le N). \tag{9}$$

Then

$$t_j(\omega_k) = \begin{cases} 0, & j \ne k, \\ t'(\omega_j) \ne 0, & j = k, \end{cases} \qquad \ell < j, k \le N, \tag{10}$$

where the nonvanishing follows from the distinctness of the tag points. Also, $t(\omega_i) \ne 0$ for $1 \le i \le \ell$.

For a witness assignment $\mathbf{w} = (a_{\ell+1}, \dots, a_N) \in \{-1, +1\}^n$, set

$$F_{\mathbf{w}}(X, Z) = \prod_{j=\ell+1}^{N} ((Z - \omega_j)X + a_j), \qquad F_{-\mathbf{w}}(X, Z) = \prod_{j=\ell+1}^{N} ((Z - \omega_j)X - a_j). \tag{11}$$

Their product is independent of the witness:

$$H(X,Z) = \prod_{j=\ell+1}^{N} \big((Z-\omega_j)X+1\big)\big((Z-\omega_j)X-1\big) = \prod_{j=\ell+1}^{N} \big((Z-\omega_j)^2X^2-1\big), \tag{12}$$

so

$$F_{\mathbf{w}}(X,Z)F_{-\mathbf{w}}(X,Z) = H(X,Z). \tag{13}$$

Writing

$$F_{\mathbf{w}}(X,Z) = \sum_{k=0}^{n} F_{\mathbf{w},k}(Z)X^k,$$

the two leading coefficients are

$$F_{\mathbf{w},n}(Z) = t(Z), \qquad F_{\mathbf{w},n-1}(Z) = \sum_{j=\ell+1}^{N} a_j t_j(Z). \tag{14}$$

Hence

$$\frac{F_{\mathbf{w},n-1}(\omega_j)}{t'(\omega_j)} = a_j \qquad (\ell < j \le N). \tag{15}$$

Lemma 5.4 (Elementary factor selection)**.** *Suppose $F, B \in \mathbb{F}[X,Z]$ satisfy*

$$F(X,Z)B(X,Z) = H(X,Z), \qquad \deg_X F = \deg_X B = n.$$

Then F is a nonzero field scalar times a product of n distinct factors from

$$\{(Z-\omega_j)X + b : \ell < j \le N,\ b \in \{-1,+1\}\}.$$

If its leading coefficient is $v_0 t(Z)$, then exactly one factor with each witness tag occurs and there is a unique $\mathbf{w} \in \{-1,+1\}^n$ such that

$$F(X,Z) = v_0 F_{\mathbf{w}}(X,Z).$$

Proof. Temporarily regard Z as symbolic and work with univariate polynomials in X whose coefficients are rational functions of Z. The $2n$ roots of H are

$$\frac{+1}{Z-\omega_j}, \qquad \frac{-1}{Z-\omega_j}, \qquad \ell < j \le N,$$

and are pairwise distinct. Here $\mathrm{char}(\mathbb{F}) > 3$ in particular implies $+1 \neq -1$; together with the distinct tag points this rules out collisions among the displayed rational functions. A degree-n factor therefore selects n of the corresponding linear factors, up to a nonzero rational scalar $v(Z)$. The constant coefficient in X of every selected product is ± 1. Since both F and the complementary factor $B = H/F$ have polynomial coefficients, both $v(Z)$ and $v(Z)^{-1}$ are polynomials; hence $v(Z)$ is a nonzero field constant.

If $c_j \in \{0,1,2\}$ is the number of selected factors carrying tag j, the leading coefficient of the selected product is $\prod_j (Z-\omega_j)^{c_j}$. Equality with $v_0 t(Z)$ says that the zero at each $Z = \omega_j$ has multiplicity one. Thus $c_j = 1$ for every witness tag, and the linear factors of F determine a unique sign vector $\mathbf{w}$. □

5.3 Public-input interpolation and targets

Assume $\ell \geq 1$; a relation with no variable public input can be given one dummy public coordinate whose domain is the singleton $\{+1\}$. For $1 \leq i \leq \ell$, let $I_i(Z) \in \mathbb{F}^{\leq \ell-1}[Z]$ be the Lagrange polynomial on the public tags:

$$I_i(Z) = \prod_{\substack{1 \leq j \leq \ell \\ j \neq i}} \frac{Z - \omega_j}{\omega_i - \omega_j}, \qquad I_i(\omega_j) = \delta_{ij}. \tag{16}$$

For $\mathbf{u} \in \mathbb{F}^\ell$, define

$$J_{\mathbf{u}}(Z) = \sum_{i=1}^{\ell} u_i I_i(Z). \tag{17}$$

The honest left and right factors are

$$\mathsf{A}_{\mathbf{u},\mathbf{w}}(X, Y, Z) = \big(Y - J_{\mathbf{u}}(Z)\big) F_{\mathbf{w}}(X, Z), \qquad \mathsf{B}_{\mathbf{w}}(X, Z) = F_{-\mathbf{w}}(X, Z). \tag{18}$$

The target family is

$$H_{\mathbf{u}}(X, Y, Z) = \big(Y - J_{\mathbf{u}}(Z)\big) H(X, Z) = Y H(X, Z) - \sum_{i=1}^{\ell} u_i I_i(Z) H(X, Z). \tag{19}$$

Thus the public input appears linearly, as required by Definition 4.1.

5.4 The two fixed spaces

Define the right triangular space

$$\mathcal{V} = \mathrm{span}_{\mathbb{F}}\{X^j Z^k : 0 \leq j \leq n,\ 0 \leq k \leq j\}. \tag{20}$$

Equivalently, $\mathsf{B} = \sum_{j=0}^{n} B_j(Z) X^j$ belongs to $\mathcal{V}$ exactly when $B_j \in \mathbb{F}^{\leq j}[Z]$. Its dimension is

$$d_V = \dim \mathcal{V} = \frac{(n+1)(n+2)}{2}. \tag{21}$$

The left space $\mathcal{U}$ will be a proper subspace of the triangular ambient space

$$\mathrm{span}_{\mathbb{F}}\Big(\{Y X^j Z^k : 0 \leq j \leq n,\ 0 \leq k \leq j\} \cup \{X^j Z^k : 0 \leq j \leq n,\ 0 \leq k \leq j + \ell - 1\}\Big).$$

Write a generic left polynomial as

$$\mathsf{A}(X, Y, Z) = Y F(X, Z) + C(X, Z), \tag{22}$$

where

$$F(X, Z) = \sum_{j=0}^{n} F_j(Z) X^j, \qquad C(X, Z) = \sum_{j=0}^{n} C_j(Z) X^j.$$

When $F_n(Z) = v_0 t(Z)$, define the encoded scaled assignment

$$\mathrm{asg}(\mathsf{A}) = \left(v_0, \left(-\frac{C_n(\omega_i)}{t(\omega_i)} \right)_{i=1}^{\ell}, \left(\frac{F_{n-1}(\omega_j)}{t'(\omega_j)} \right)_{j=\ell+1}^{N} \right) \in \mathbb{F}^{N+1}. \tag{23}$$

All denominators are fixed nonzero field elements, so this is linear in the coefficient vector once the leading condition is imposed.

Define $\mathcal{U}$ to consist of all polynomials $\mathsf{A} = YF + C$ satisfying

$$F_j \in \mathbb{F}^{\le j}[Z] \qquad (0 \le j \le n), \tag{24}$$

$$C_j \in \mathbb{F}^{\le j+\ell-1}[Z] \qquad (0 \le j \le n), \tag{25}$$

$$F_n(Z) = v_0 t(Z) \qquad \text{for some } v_0 \in \mathbb{F}, \tag{26}$$

$$\mathrm{asg}(\mathsf{A})D = \mathbf{0}. \tag{27}$$

Every condition is homogeneous and linear in the coefficients. The ambient left space before the fixed constraints has dimension

$$\sum_{j=0}^{n}(j+1) + \sum_{j=0}^{n}(j+\ell) = (n+1)(N+1). \tag{28}$$

The leading condition has codimension n. After imposing it, the assignment map is surjective: v_0 is chosen through F_n, the degree bound on C_n permits arbitrary values at the ℓ public tag points, and the degree bound on F_{n-1} permits arbitrary values at the n witness tag points. These choices use independent coefficient blocks. The kernel condition therefore has codimension $\mathrm{rank}(D)$, and

$$d_U = \dim \mathcal{U} = (n+1)(N+1) - n - \mathrm{rank}(D). \tag{29}$$

For the circuit matrix constructed earlier, $\mathrm{rank}(D) = G$.

Remark 5.5 (Constants and affine proof maps)**.** The constant polynomial 1 lies in both spaces. For $\mathcal{U}$, take $F = 0$ and $C = 1$, so $v_0 = 0$ and $\mathrm{asg}(1) = (0, \ldots, 0) \in \mathbb{F}^{N+1}$. Bases may therefore begin with 1, and affine proof maps need no separate CRS coordinates or multiplicative tags.

Lemma 5.6 (Honest membership)**.** *If $(\mathbf{u}, \mathbf{w}) \in \mathcal{R}$, then*

$$\mathsf{A}_{\mathbf{u},\mathbf{w}} \in \mathcal{U}, \qquad \mathsf{B}_{\mathbf{w}} \in \mathcal{V}.$$

Proof. The triangular degree bounds follow from multiplying the witness linear factors. Equations (14) and (17) give

$$\mathrm{asg}(\mathsf{A}_{\mathbf{u},\mathbf{w}}) = \overline{\mathbf{a}}(\mathbf{u}, \mathbf{w}),$$

so the circuit constraint follows from the relation. The complementary factor satisfies the same triangular degree bounds and belongs to $\mathcal{V}$. □

5.5 Exact factorization realizes the signed relation

Theorem 5.7 (Realization theorem)**.** *The factor map in* (18) *and the coefficient decoder below make the concrete FSP realize $\mathcal{R}$.*

Proof. The encoding direction follows from Lemma 5.6 and (13). Conversely, suppose

$$\mathsf{A}(X, Y, Z)\mathsf{B}(X, Z) = H_{\mathbf{u}}(X, Y, Z) = \big(Y - J_{\mathbf{u}}(Z)\big)H(X, Z)$$

with $\mathsf{A} = YF + C \in \mathcal{U}$ and $\mathsf{B} \in \mathcal{V}$. Comparing coefficients of Y gives

$$F(X, Z)\mathsf{B}(X, Z) = H(X, Z). \tag{30}$$

The constant coefficient gives

$$C(X,Z)\mathsf{B}(X,Z) = -J_{\mathbf{u}}(Z)H(X,Z) = -J_{\mathbf{u}}(Z)F(X,Z)\mathsf{B}(X,Z).$$

Since $\mathsf{B} \neq 0$,

$$C(X,Z) = -J_{\mathbf{u}}(Z)F(X,Z). \tag{31}$$

Both factors in (30) have X-degree at most n, whereas H has degree $2n$, so both have degree exactly n. The leading condition in $\mathcal{U}$ has $F_n = v_0 t$ with $v_0 \neq 0$. Lemma 5.4 therefore gives a unique $\mathbf{w} \in \{-1,+1\}^n$ with $F = v_0 F_{\mathbf{w}}$. Equation (31) then yields

$$\mathrm{asg}(\mathsf{A}) = v_0 \overline{\mathbf{a}}(\mathbf{u},\mathbf{w}).$$

The fixed kernel constraint in $\mathcal{U}$ implies $v_0\overline{\mathbf{a}}(\mathbf{u},\mathbf{w})D = 0$, and $v_0 \neq 0$ gives $(\mathbf{u},\mathbf{w}) \in \mathcal{R}$.

The decoder reads v_0 from $F_n = v_0 t$ and outputs

$$a_j = \frac{F_{n-1}(\omega_j)}{v_0 t'(\omega_j)} \qquad (\ell < j \le N), \tag{32}$$

followed by explicit sign and matrix checks. □

5.6 Every public target lies in the product span

Lemma 5.8 (Triangular residual support)**.** *Let $E(Z) \in \mathbb{F}^{\le n-1}[Z]$ and set*

$$F_E(X,Z) = t(Z)X^n + E(Z)X^{n-1}, \qquad B_E(X,Z) = t(Z)X^n - E(Z)X^{n-1}.$$

Every monomial $X^d Z^e$ in

$$K_E(X,Z) = H(X,Z) - F_E(X,Z)B_E(X,Z)$$

satisfies

$$0 \le d \le 2n-2, \qquad 0 \le e \le d.$$

Proof. In $H(X,Z) = \prod_j ((Z-\omega_j)^2X^2 - 1)$, every occurrence of X^2 contributes a polynomial of Z-degree at most two. Hence every monomial X^dZ^e of H satisfies $e \le d$. Moreover,

$$F_E B_E = t(Z)^2 X^{2n} - E(Z)^2 X^{2n-2}.$$

The leading term cancels the X^{2n} term of H, and both polynomials have no X^{2n-1} term. Finally, $\deg E^2 \le 2n-2$, so the additional X^{2n-2} term also satisfies the claimed triangular bound. □

Theorem 5.9 (Target-span theorem)**.** *Assume $n \ge 2$. For every $\mathbf{u} \in \mathbb{F}^\ell$,*

$$H_{\mathbf{u}}(X,Y,Z) \in \mathrm{span}(\mathcal{U}\mathcal{V}).$$

Moreover, for fixed bases $\mathbf{U},\mathbf{V}$ there are efficiently computable fixed matrices $M_0, M_1, \ldots, M_\ell$ satisfying

$$\mathbf{U}M_0\mathbf{V}^T = YH, \tag{33}$$

$$\mathbf{U}M_i\mathbf{V}^T = -I_i H \qquad (1 \le i \le \ell), \tag{34}$$

and hence

$$M_{\mathbf{u}} = M_0 + \sum_{i=1}^{\ell} u_i M_i \tag{35}$$

represents $H_{\mathbf{u}}$.

Proof. Fix $\mathbf{u} \in \mathbb{F}^\ell$ and use Lemma 5.2 to compute $\mathbf{w}^\star = \mathbf{w}^\star(\mathbf{u}) \in \mathbb{F}^n$ with $\overline{\mathbf{a}}(\mathbf{u}, \mathbf{w}^\star)D = 0$. Define

$$E^\star(Z) = \sum_{j=\ell+1}^{N} a_j^\star t_j(Z),$$

$$F^\star(X,Z) = t(Z)X^n + E^\star(Z)X^{n-1}, \qquad B^\star(X,Z) = t(Z)X^n - E^\star(Z)X^{n-1},$$

and

$$\mathsf{A}^\star(X,Y,Z) = \big(Y - J_{\mathbf{u}}(Z)\big)F^\star(X,Z).$$

The assignment map gives $\mathrm{asg}(\mathsf{A}^\star) = \overline{\mathbf{a}}(\mathbf{u}, \mathbf{w}^\star)$, so $\mathsf{A}^\star \in \mathcal{U}$; also $B^\star \in \mathcal{V}$.

By Lemma 5.8, the polynomial

$$K^\star(X,Z) = H(X,Z) - F^\star(X,Z)B^\star(X,Z)$$

has only monomials $X^d Z^e$ with $0 \le d \le 2n-2$ and $0 \le e \le d$. Furthermore,

$$\Delta_{\mathbf{u}} := H_{\mathbf{u}} - \mathsf{A}^\star B^\star = \big(Y - J_{\mathbf{u}}(Z)\big)K^\star(X,Z).$$

Consequently, a Y-monomial in $\Delta_{\mathbf{u}}$ has the form YX^dZ^e with $0 \le d \le 2n-2$ and $0 \le e \le d$. Set $i = \max(0, d-n)$ and $j = d - i$. Then $0 \le i \le n-2$ and $0 \le j \le n$. Recall that the additional restrictions defining $\mathcal{U}$ involve only F_n, C_n, and F_{n-1}, hence only the X^n and X^{n-1} coefficient blocks. Since $i \le n-2$, every triangular monomial of left X-degree i used below belongs to $\mathcal{U}$. Set $e_L = \max\{0, e-j\}$ and $e_R = e - e_L$. Then $e_L \le i$ and $e_R \le j$.

$$YX^dZ^e = (YX^iZ^{e_L})(X^jZ^{e_R}),$$

where the left factor is an unrestricted lower F_i monomial in $\mathcal{U}$ and the right factor lies in $\mathcal{V}$.

A Y-independent monomial has the form X^dZ^e with the same range of d and $0 \le e \le d + \ell - 1$. Set $e_L = \max\{0, e-j\}$ and $e_R = e - e_L$. Then $e_L \le i + \ell - 1$ and $e_R \le j$.

$$X^dZ^e = (X^iZ^{e_L})(X^jZ^{e_R}),$$

where the left factor is an unrestricted lower C_i monomial in $\mathcal{U}$. Thus every monomial of $\Delta_{\mathbf{u}}$ belongs to $\mathrm{span}(\mathcal{U}\mathcal{V})$, proving the first claim.

Setting $\mathbf{u} = \mathbf{0}$ gives $H_{\mathbf{0}} = H_0 = YH$, and hence a product-span representation of YH. For $1 \le i \le \ell$, let $\mathbf{e}_i \in \mathbb{F}^\ell$ denote the ith standard basis vector. By the definition of the target family,

$$H_{\mathbf{e}_i} = YH - I_iH,$$

so subtracting a representation of $H_{\mathbf{e}_i}$ from the representation of YH gives a representation of I_iH. Negating this representation and writing all representations in the fixed bases $\mathbf{U}$ and $\mathbf{V}$ gives the claimed matrices $M_0, M_1, \ldots, M_\ell$. □

Theorem 5.10 (Concrete FSP)**.** *The spaces $\mathcal{U}, \mathcal{V}$, the target components*

$$H_0(X,Y,Z) = YH(X,Z), \qquad H_i(X,Y,Z) = -I_i(Z)H(X,Z),$$

and the matrices $M_0, M_1, \ldots, M_\ell$ from Theorem 5.9 form an input-linear FSP realizing $\mathcal{R}$. Its dimensions satisfy

$$d_U = (n+1)(N+1) - n - \mathrm{rank}(D), \qquad d_V = \frac{(n+1)(n+2)}{2},$$

and its degree parameter is bounded by

$$d_{\mathsf{FSP}} \le 4n + \max\{1, \ell - 1\} \le 4N + 1. \tag{36}$$

Proof. Realization is Theorem 5.7, and target representability is Theorem 5.9. The dimensions were computed above. The triangular bounds give

$$\deg_{\mathrm{tot}}(YF) \le 2n+1, \qquad \deg_{\mathrm{tot}} C \le 2n+\ell-1, \qquad \deg_{\mathrm{tot}} \mathsf{B} \le 2n.$$

Hence

$$\deg_{\mathrm{tot}}(\mathsf{AB}) \le \max\{4n+1, 4n+\ell-1\} = 4n + \max\{1, \ell-1\}.$$

The same bound covers the target components YH and I_iH. □

6 The split NILP and pairing argument

The field-level proof system is now the compiler of Theorem 4.4 applied to Theorem 5.10. We spell it out to make the split structure and the short verifier explicit.

6.1 Field-level NILP algorithms

Choose bases

$$\mathbf{U}(X,Y,Z) = (U_1, \dots, U_{d_U}), \qquad \mathbf{V}(X,Z) = (V_1, \dots, V_{d_V})$$

with $U_1 = V_1 = 1$. Instantiating the syntax above, $\mathsf{Setup}(\mathcal{R})$ samples independently

$$x, y, z \leftarrow \mathbb{F}$$

and computes

$$\boldsymbol{\sigma}_1 = \mathbf{U}(x,y,z), \qquad \boldsymbol{\sigma}_2 = \mathbf{V}(x,z), \qquad \mathsf{td} = (x,y,z). \tag{37}$$

On input $(\mathbf{u}, \mathbf{w}) \in \mathcal{R}$, the algorithm $\mathsf{ProofMatrix}$ samples $r \leftarrow \mathbb{F}^\times$ and forms

$$\widetilde{\mathsf{A}} = r\mathsf{A}_{\mathbf{u},\mathbf{w}}, \qquad \widetilde{\mathsf{B}} = r^{-1}\mathsf{B}_{\mathbf{w}}. \tag{38}$$

Let $\mathbf{c}_A \in \mathbb{F}^{d_U}$ and $\mathbf{c}_B \in \mathbb{F}^{d_V}$ be their unique coefficient rows in the fixed bases, so

$$\widetilde{\mathsf{A}} = \mathbf{c}_A \mathbf{U}^T, \qquad \widetilde{\mathsf{B}} = \mathbf{c}_B \mathbf{V}^T.$$

It returns the two proof matrices

$$P_1 = \mathbf{c}_A \in \mathbb{F}^{1\times d_U}, \qquad P_2 = \mathbf{c}_B \in \mathbb{F}^{1\times d_V}.$$

Thus $\mathsf{ProofMatrix}$ depends on $(\mathbf{u}, \mathbf{w}, r)$ but not on the sampled evaluation point (x,y,z) or the CRS values. The prover algorithm

$$\mathsf{Prove}(\mathcal{R}, \boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \mathbf{u}, \mathbf{w})$$

runs $\mathsf{ProofMatrix}(\mathcal{R}, \mathbf{u}, \mathbf{w})$, applies the resulting matrices to the corresponding CRS halves, and outputs

$$\mathsf{proof} = (A, B), \qquad A = \boldsymbol{\sigma}_1 P_1^T = \widetilde{\mathsf{A}}(x,y,z), \qquad B = \boldsymbol{\sigma}_2 P_2^T = \widetilde{\mathsf{B}}(x,z). \tag{39}$$

For an instance $\mathbf{u}$, the algorithm Test returns the single matrix

$$T_{\mathbf{u}} = \begin{pmatrix} M_{\mathbf{u}} & 0 \\ 0 & -1 \end{pmatrix}, \qquad M_{\mathbf{u}} = M_0 + \sum_{i=1}^{\ell} u_i M_i,$$

where the zero blocks have dimensions $d_U \times 1$ and $1 \times d_V$. On input $(\boldsymbol{\sigma}_1, \boldsymbol{\sigma}_2, \mathbf{u}, \mathsf{proof})$, the verifier algorithm Vfy rejects unless every $u_i \in \{-1, +1\}$ and otherwise accepts precisely when

$$(\boldsymbol{\sigma}_1, A) T_{\mathbf{u}} (\boldsymbol{\sigma}_2, B)^T = 0, \qquad \text{equivalently} \qquad AB = \boldsymbol{\sigma}_1 M_{\mathbf{u}} \boldsymbol{\sigma}_2^T. \tag{40}$$

Each proof matrix has one row, so the proof has split dimensions $(1,1)$; the verifier evaluates one cross-bilinear equation.

Corollary 6.1 (Field-level security)**.** *The NILP has perfect completeness, split-affine statistical knowledge error at most d_{FSP}/q, and disclosure disagreement at most d_{FSP}/q.*

Proof. Apply Theorem 4.4. The decoder is given explicitly in (32). □

6.2 Compilation to Type-III pairings

We use Groth's generic bilinear-group compiler [Gro16, Section 2.5] to derive a preprocessing zk-SNARK from the $(1, 1)$-split NILP.

Let

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_T$$

be a Type-III bilinear map whose three groups have prime order q, and take $\mathbb{F} = \mathbb{F}_q$. We use additive notation for all three groups. Fix generators $G_1 \in \mathbb{G}_1$ and $G_2 \in \mathbb{G}_2$, and let $G_T = e(G_1, G_2) \in \mathbb{G}_T$. Write $[a]_1 = aG_1$, $[a]_2 = aG_2$, and $[a]_T = aG_T$, so $e([a]_1, [b]_2) = [ab]_T$.

Encode the two CRS halves as

$$\boldsymbol{\sigma}_1 = [\mathbf{U}(x, y, z)]_1, \qquad \boldsymbol{\sigma}_2 = [\mathbf{V}(x, z)]_2. \tag{41}$$

The proof becomes

$$\mathsf{proof} = ([A]_1, [B]_2) \in \mathbb{G}_1 \times \mathbb{G}_2, \tag{42}$$

which can be computed linearly from the CRS halves using the proof matrices as in (39).

A full verifier can evaluate the right-hand side of (40) by a pairing-product computation. To get a fast online verifier, we can precompute

$$\boldsymbol{\sigma}_{\mathrm{V}} = (K_0, K_1, \ldots, K_\ell), \tag{43}$$

where

$$K_0 = [yH(x, z)]_T, \qquad K_i = [-I_i(z)H(x, z)]_T. \tag{44}$$

These elements can also be computed directly from (x, y, z) during the setup without materializing the representation matrices $M_0, \ldots, M_\ell$. By Theorem 5.9, the same elements are derivable from the prover CRS using those fixed matrices, so this does not change the security guarantees of the proof system.

Verification therefore becomes

$$e([A]_1, [B]_2) \stackrel{?}{=} K_0 + \sum_{i=1}^{\ell} u_i K_i. \tag{45}$$

The online verifier uses one pairing and ℓ target-group additions or subtractions, independent of N.

Groth's disclosure-free split-NILP compiler [Gro16, Section 2.5] gives the compiled argument perfect completeness and statistical knowledge soundness against adversaries making polynomially many generic bilinear-group operations. To interpret the statement asymptotically, consider a family indexed by a security parameter λ in which $N = N(\lambda)$ is polynomially bounded, $1/q(\lambda)$ is negligible, and $\log q(\lambda) = \mathrm{poly}(\lambda)$. Since $d_{\mathsf{FSP}} \le 4N + 1$, the field-level knowledge and disclosure errors are then negligible, as required by the compiler theorem.

6.3 Rerandomization and statistical zero knowledge

The polynomial and proof pairs admit the public rerandomization

$$(\mathsf{A}, \mathsf{B}) \longmapsto (r\mathsf{A}, r^{-1}\mathsf{B}), \qquad (A, B) \longmapsto (rA, r^{-1}B), \qquad r \leftarrow \mathbb{F}^{\times}. \tag{46}$$

The same map is applied by scalar multiplication after group compilation.

Theorem 6.2 (Statistical composable zero knowledge)**.** *With rerandomization, the NILP satisfies Definition 3.2 with statistical composable zero-knowledge error*

$$\zeta < \frac{2n}{q}.$$

Its simulated setup samples the real setup distribution conditioned on $H(x, z) \neq 0$, and proof simulation under the resulting CRS is perfect even given the simulation trapdoor.

Proof. Let

$$\rho = \Pr_{x,z \leftarrow \mathbb{F}}[H(x, z) = 0]. \tag{47}$$

For a fixed witness tag j, one of its two factors in H vanishes exactly when $(z - \omega_j)x \in \{-1, +1\}$. For each $z \neq \omega_j$ there are exactly two values of x, and for $z = \omega_j$ there are none. A union bound gives

$$\rho \leq \frac{2n(q-1)}{q^2} < \frac{2n}{q}. \tag{48}$$

The algorithm $\mathsf{SimSetup}$ rejection-samples $x, y, z \leftarrow \mathbb{F}$ until $H(x, z) \neq 0$, constructs the same CRS as Setup, and retains $\widehat{\mathsf{td}} = (x, y, z)$. The statistical distance between the uniform distribution of (x, y, z) and this conditioned distribution is exactly ρ. Since the CRS is a deterministic function of these values,

$$\Delta(\mathsf{CRS}, \widehat{\mathsf{CRS}}) \leq \rho < \frac{2n}{q}.$$

It remains to prove perfect simulation under the simulated CRS. Let $T = (y - J_{\mathbf{u}}(z))H(x, z)$. If $y \neq J_{\mathbf{u}}(z)$, the simulator samples $A \leftarrow \mathbb{F}^{\times}$ uniformly and sets $B = T/A$. This is exactly the honest rerandomized proof distribution. If $y = J_{\mathbf{u}}(z)$, the honest left proof is always $A = 0$ and the right proof is uniform in $\mathbb{F}^{\times}$; the simulator outputs $(0, B)$ for uniform $B \in \mathbb{F}^{\times}$. The distribution is therefore identical for every valid adaptively chosen instance–witness pair, without the simulator using the witness. The equality holds pointwise for every simulated trapdoor, so it continues to hold when the environment is given that trapdoor. This proves perfect proof simulation under $\mathsf{SimSetup}$ and the claimed statistical composable zero-knowledge error. □

Remark 6.3 (Efficiency of simulated setup)**.** The simulated setup is efficient throughout the parameter range of Theorem 8.1. If $q > 2n$, rejection sampling takes fewer than $(1 - 2n/q)^{-1} \leq 2n + 1$ trials in expectation. If $n < q \leq 2n$, the choice $x = 0$ always satisfies $H(x, z) \neq 0$, so a trial succeeds with probability at least $1/q$ and the expected number of trials is at most $q \leq 2n$. This conditioning affects only the simulator: the real setup remains uniform, so the knowledge and disclosure bounds proved for it are unchanged. The pairing compiler preserves perfect simulation under the simulated CRS by encoding the two field elements in the corresponding source groups.

Object	Size or cost
Nonconstant signed coordinates	$N = W + G$
Witness coordinates	$n = N - \ell$
Left prover CRS	$\dim \mathcal{U} = (n+1)(N+1) - n - G$ field or $\mathbb{G}_1$ elements
Right prover CRS	$\dim \mathcal{V} = (n+1)(n+2)/2$ field or $\mathbb{G}_2$ elements
Proof	two field elements; after compilation, one element in each source group
Verifier key	$\ell + 1$ target-group elements
Online verifier	one pairing and $O(\ell)$ additions/subtractions in $\mathbb{G}_T$
Field-level knowledge error	at most d_{FSP}/q, where $d_{\mathsf{FSP}} \le 4N + 1$
Disclosure disagreement	at most d_{FSP}/q
Composable statistical ZK error	$\zeta < 2n/q$ after rerandomization

Table 1: Concrete costs for the circuit-specialized construction. The displayed formula $d_U = (n+1)(N+1) - n - G$ uses the full-rank circuit matrix, for which $\operatorname{rank}(D) = G$.

6.4 Efficiency

Table 1 records the costs for a circuit-specialized setup. As in QAP-based preprocessing, the prover CRS depends on the fixed circuit and on which wires are public, but it can be reused for every instance with that public–witness partition.

The table counts encoded CRS elements, not a dense representation of every fixed linear map in the FSP abstraction. Since $d_U, d_V = O(N^2)$, one dense $d_U \times d_V$ matrix M_i may contain $O(N^4)$ field elements, and the entire family may contain $O(N^5)$ when $\ell = O(N)$. These matrices witness product-span membership in the abstract FSP and need not be materialized by the concrete pairing setup. After sampling (x, y, z), setup evaluates $H_0(x, y, z), \ldots, H_\ell(x, y, z)$ directly to obtain the target-group verifier key in (44).

There remains fixed basis preprocessing for $\mathcal{U}$. Its ambient coefficient space has $O(N^2)$ coordinates and its defining system has $O(N)$ independent linear constraints. A dense systematic row reduction therefore gives a conservative $O(N^4)$-time and $O(N^3)$-working-memory preprocessing bound. The resulting systematic basis has $O(N^3)$ potentially nonzero field coefficients, while conversion of a valid polynomial coefficient table to basis coordinates consists simply of reading its free coordinates. These fixed representation costs are separate from the $O(N^2)$ encoded prover CRS size displayed in Table 1.

The right prover CRS has exactly

$$d_V = \frac{(n+1)(n+2)}{2}$$

source-group elements. For the circuit matrix, $\operatorname{rank}(D) = G$, so the left prover CRS has exactly

$$d_U = (n+1)(N+1) - n - G$$

elements.

Each selected or complementary factor polynomial has $d_V = (n+1)(n+2)/2$ triangular coefficients. The polynomial $C = -J_{\mathbf{u}} F_{\mathbf{w}}$ has $\sum_{j=0}^{n} (j + \ell)$ coefficient slots. After multiplying k tagged factors, the triangular coefficient table has $\Theta(k^2)$ entries. Multiplication by one further three-monomial factor therefore costs $O(k^2)$ field operations, and sequential multiplication costs $\sum_{k=1}^{n} O(k^2) = O(n^3)$. A balanced product tree using standard fast dense bivariate multiplication

reduces this to $\widetilde{O}(n^2)$ field operations, quasi-linear in the $\Theta(n^2)$ output table, up to the usual polylogarithmic factors.

The right space uses its monomial basis. For the left space, setup uses the systematic row-echelon basis described above. A valid polynomial coefficient table is converted to basis coordinates by reading its free coordinates, not by solving an instance-dependent linear system. The compiled prover performs one multi-scalar multiplication of size d_U in $\mathbb{G}_1$ and one of size d_V in $\mathbb{G}_2$.

Verification checks the public signs, evaluates one pairing, performs ℓ additions or subtractions in $\mathbb{G}_T$, and tests one equality. There is no online verifier dependence on N when the target group elements $K_0, K_1, \ldots, K_\ell$ are precomputed.

7 Related work

7.1 Boolean constraints, span programs, and signed linear systems

Groth, Ostrovsky, and Sahai observed (Section 4.2, Lemma 3) that, for bits $b_0, b_1, b_2 \in \{0, 1\}$,

$$b_0 + b_1 + 2b_2 - 2 \in \{0, 1\} \quad \Longleftrightarrow \quad b_2 = b_0 \,\mathsf{NAND}\, b_1,$$

and used homomorphic commitments to prove that the residual is Boolean [GOS06]. The signed NAND constraint in Section 2 is this observation after the change of variables $a = 2b - 1$ and the introduction of an explicit residual sign. There is also a half-adder interpretation: for a valid NAND gate, $1 - b_2 = b_0 b_1$ is the carry and $b_0 + b_1 + 2b_2 - 2 = b_0 \oplus b_1$ is the sum.

Danezis, Fournet, Groth, and Kohlweiss [DFGK14] characterize all nontrivial fan-in-two Boolean gates through affine maps whose values are restricted to signed coordinates. The system $\overline{\mathbf{a}}D = 0$ is a homogeneous residual-variable form of a similar affine-map characterization. They reduce the relation to a square span program (SSP), a notion defined in the same paper, and use pairings to obtain a SNARK. Our construction instead uses a factorization span program.

Karchmer and Wigderson introduced span programs as a linear-algebraic model of Boolean computation [KW93]. In a standard span program, the input assignment determines which labelled vectors are available and acceptance asks whether a fixed target belongs to their span; the coefficients of the spanning combination are otherwise arbitrary field elements. The signed linear system used here has the complementary form: the matrix D is fixed, while the assignment is itself the coefficient vector, constrained coordinatewise to $\{-1, +1\}$, and is required to lie in the left kernel of D.

Quadratic span programs (QSPs) and quadratic arithmetic programs (QAPs) provide the main precedent for relation-specific preprocessing and for compressing many constraints into one polynomial condition [GGPR13]. Square arithmetic programs (SAPs) exploit the corresponding one-polynomial symmetry in the arithmetic setting [GM17]. Factorization span programs instead distinguish membership in a product span from the existence of a single exact factorization.

Lipmaa gives a unified framework for non-universal SNARKs encompassing QAP, SAP, SSP, and QSP constructions and relates several of their underlying languages [Lip22]. FSP acceptance has a different defining condition: the public target must have a decomposable preimage, rather than merely satisfying the quadratic or square-program condition of those frameworks.

Polynomial products have also appeared before as proof constraints. Neff's shuffle argument uses the invariance of a monic polynomial under permutation of its roots to certify multiset equality [Nef01]. Our use of products is different: the contribution is the complementary tagged-factor FSP and its split NILP, not the general use of polynomial factorization in zero knowledge.

The conscientious span programs of Gennaro, Gentry, Parno, and Raykova ensure that every input to a gate checker is actually used, so that wire values can be extracted consistently [GGPR13].

Lipmaa subsequently gave a modular circuit checker that combines small standard span programs for individual gates with a high-distance linear error-correcting code enforcing consistency between wire occurrences [Lip13]. In the present formulation, wire coordinates are shared globally and every gate has a dedicated residual coordinate. Gate correctness and wire consistency are thereby expressed by one sparse homogeneous linear system, without a separate wire checker.

There is also a direct partition interpretation of this system. If $\mathbf{d}_0, \ldots, \mathbf{d}_N \in \mathbb{F}^G$ are the rows of D, then $\overline{\mathbf{a}}D = 0$ is the signed vector-partition condition

$$\sum_{i=0}^{N} a_i \mathbf{d}_i = 0, \qquad a_i \in \{-1, +1\}.$$

Equivalently, the rows assigned sign $+1$ and those assigned sign -1 have equal sums. Under $x_i = (a_i + 1)/2$, this becomes the binary linear-feasibility equation $\mathbf{x}D = \frac{1}{2}\mathbf{1}D$, equivalently a multidimensional subset-sum equation. Fauzi, Lipmaa, and Zhang constructed a NIZK argument for the scalar Set-Partition relation $\sum_i b_i s_i = 0$ with $b_i \in \{-1, +1\}$ [FLZ13]. Our relation is a sparse vector-valued version with a distinguished constant coordinate and partially fixed public signs; the proof mechanism is different.

7.2 From LIPs to split NILPs

Bitansky, Chiesa, Ishai, Ostrovsky, and Paneth introduced linear interactive proofs to isolate the information-theoretic linear-prover object behind several preprocessing SNARKs [BCI+13]. Groth recast their particular two-move algebraic input-oblivious LIP as a NILP, placing the verifier's first message in setup and writing the affine prover strategy as a proof matrix [Gro16]. For that restricted LIP notion, NILP is therefore largely a change of presentation, but it is not synonymous with LIP in general.

Dot-product proofs provide another information-theoretic linear proof object: the verifier makes one fully linear query jointly to the instance and proof. Their generic-group applications yield very short interactive arguments, but do not give a publicly verifiable noninteractive Type-III argument with one element from each source group [BHI+24].

The split-NILP definition is a substantive refinement for Type-III pairings: the CRS and proof are partitioned into left and right parts, each proof part may depend only on its corresponding setup half, and verification is cross-bilinear. Two-field-element NILPs already follow from SSPs or squaring-only circuits. The open issue is whether the responses can be separated as one element from each Type-III side. The contribution is therefore not the first two-field-element NILP, but an FSP and resulting disclosure-free NILP with split dimensions exactly $(1, 1)$.

The split-NILP perspective has also been used explicitly after Groth. Groth and Maller formulate pairing-based simulation-extractable arguments in terms of split NILPs [GM17], while Campanelli, Fiore, and Querol restate the split-NILP compiler and use variants of Groth's NILP to obtain modular commit-and-prove SNARKs [CFQ19]. These works retain at least three proof elements in the Type-III setting and do not give split dimensions $(1, 1)$.

7.3 Short pairing-based arguments

The progression toward short pairing-based arguments includes Groth's initial constant-size construction, Lipmaa's refinement, the seven-element QSP argument, the four-element SSP argument, and Groth's three-element argument [Gro10, Lip12, GGPR13, DFGK14, Gro16]. The last remains the principal Type-III pairing baseline without a random oracle.

Two proof elements were already attainable from squaring-only arithmetizations using symmetric Type-I pairings [Gro16]. This does not yield one element in each of two distinct source groups and, in standard pairing-friendly instantiations, the larger symmetric source groups typically make the encoded proof longer at comparable security. The question left by Groth is specifically whether two elements suffice over asymmetric Type-III pairings.

Groth and Maller prove that any pairing-based simulation-extractable SNARK or signature of knowledge in their model requires at least three group elements and two verification equations [GM17]. Their lower bound concerns the stronger simulation-extractable notion and therefore does not rule out the ordinary knowledge-sound $(1, 1)$ construction given here.

Among publicly verifiable systems using a random oracle, Polymath uses KZG commitments for SAPs and, after Fiat–Shamir, has a proof consisting of three $\mathbb{G}_1$ elements and one field element, with no $\mathbb{G}_2$ proof element (see Table 1 of [Lip24]); the construction uses KZG commitments [KZG10]. PARI sends two $\mathbb{G}_1$ elements and two field elements; its noninteractive form also uses the random-oracle model [DMS26]. A later extension adds zero knowledge without changing this basic proof shape [KPS26].

Arnon, Dujmović, and Yogev give a public SNARG with exactly two $\mathbb{G}_1$ elements and no additional bits in the combined generic bilinear-group and random-oracle model [ADY26]. This is the closest result in exact group-element count. Because both elements are in $\mathbb{G}_1$, their proof can also be shorter in encoded bits on common Type-III curves. Their theorem is in a different model: the present construction addresses the plain split-NILP question without a random oracle and has perfect completeness.

7.4 Short arguments in other models

Shorter designated verifier arguments exist. Barta, Ishai, Ostrovsky, and Wu give a designated-verifier SNARG in the pairing-free generic-group model with two proof elements and inverse-polynomial soundness error [BIOW20]. Arnon, Dujmović, and Ishai later obtain negligible soundness with one group element and additional security-parameter-length bits, again for a designated verifier [ADI25]. These constructions rely on a secret verification key and do not address the publicly verifiable Type-III problem.

Sahai and Waters give a qualitatively different route to short arguments from indistinguishability obfuscation and puncturable pseudorandom functions [SW14]. This is a separate feasibility route based on incomparable assumptions and proof techniques.

8 Conclusion

Theorem 8.1 (Field-level main theorem)**.** *Let $\ell \geq 1$, $n \geq 2$, and $N = n+\ell$. Let $\mathbb{F}$ be a finite field of size $q > N$ and characteristic greater than 3, and let $D \in \mathbb{F}^{(N+1)\times G}$. Suppose the signed homogeneous system defined by D is publicly extendable in the sense of Definition 5.1. The construction gives an input-linear FSP realizing $\mathcal{R}$ and, by random evaluation, a $(1, 1)$-split NILP with:*

(i) perfect completeness;

(ii) split-affine statistical knowledge error at most d_{FSP}/q, where $d_{\mathsf{FSP}} \leq 4n+\max\{1, \ell-1\} \leq 4N+1$;

(iii) disclosure disagreement at most d_{FSP}/q;

(iv) statistical composable zero knowledge with error $\zeta < 2n/q$ under multiplicative rerandomization;

(v) prover CRS size

$$d_U + d_V = (n+1)(N+1) - n - \operatorname{rank}(D) + \frac{(n+1)(n+2)}{2},$$

one field response from each split side, and an input-linear target family whose $\ell+1$ component polynomials admit fixed representation matrices. The concrete pairing setup need not materialize these matrices.

Under the minimal condition $q > N$, the displayed algebraic error bounds may be vacuous for small fields. They are intended to become negligible in the asymptotic instantiation below.

Corollary 8.2 (Boolean circuits)**.** *Let $\mathcal{C}$ be a Boolean circuit with G gates, $\ell \geq 1$ public inputs, and W input and internal wires after eliminating the accepting output wire, and suppose $n := G + W - \ell \geq 2$. Over any field satisfying the hypotheses of Theorem 8.1, the construction gives a disclosure-free $(1,1)$-split NILP for $\mathcal{R}_{\mathcal{C}}$ with the same properties, where $N = G + W$ and $\operatorname{rank}(D) = G$.*

Corollary 8.3 (Type-III pairing compilation)**.** *Assume in addition that q is prime and that Type-III bilinear groups of order q are available. For a family indexed by a security parameter λ, suppose also that $N = N(\lambda)$ is polynomially bounded and $1/q(\lambda)$ is negligible, while $\log q(\lambda) = \operatorname{poly}(\lambda)$. Encoding the two field-level CRS halves and proof responses in the corresponding source groups gives a pairing argument whose proof consists of one $\mathbb{G}_1$ element and one $\mathbb{G}_2$ element. The derived verifier key contains $\ell+1$ target-group elements, and online verification uses one pairing and a linear combination in the public input. The argument has perfect completeness, preserves the field-level statistical composable zero knowledge with negligible error, and, by Groth's split-NILP compiler, is statistically knowledge sound against adversaries making polynomially many generic bilinear-group operations.*

The main result closes the split-dimension gap at the NILP level. Through Groth's compiler, it also answers the corresponding generic Type-III pairing question.

AI declaration

AI was used extensively in this research. Prompting GPT 5.6 Sol Pro (August 2026) with the open question yielded a split NILP for an exact knapsack problem leveraging the core ideas in the FSP. Further prompting yielded a split NILP for Boolean circuit satisfiability, and subsequently reduced the verification cost from $O(N^2)$ to $O(\ell)$. It is still an open question whether it is possible to keep the advantages of the proof system but reduce the CRS size to $O(N)$. Sol Pro found a complicated way to reduce the size to $O(N^2/\log N)$ but did not make further progress.

References

[ADI25] Gal Arnon, Jesko Dujmović, and Yuval Ishai. Designated-Verifier SNARGs with One Group Element. In *CRYPTO 2025, Part VII*, LNCS 16006, pages 192–224, 2025. `https://doi.org/10.1007/978-3-032-01907-3_7`.

[ADY26] Gal Arnon, Jesko Dujmović, and Eylon Yogev. Pairing-Based SNARGs with Two Group Elements. In *CRYPTO 2026, Part IX*, LNCS 16808, pages 125–156, 2026. Cryptology ePrint Archive, Paper 2025/2160. `https://eprint.iacr.org/2025/2160`.

[BIOW20] Ohad Barta, Yuval Ishai, Rafail Ostrovsky, and David J. Wu. On Succinct Arguments and Witness Encryption from Groups. In *CRYPTO 2020, Part I*, LNCS 12170, pages 776–806, 2020. https://eprint.iacr.org/2020/1319.

[BCI+13] Nir Bitansky, Alessandro Chiesa, Yuval Ishai, Rafail Ostrovsky, and Omer Paneth. Succinct Non-Interactive Arguments via Linear Interactive Proofs. In *TCC 2013*, LNCS 7785, pages 315–333, 2013. https://eprint.iacr.org/2012/718.

[BHI+24] Nir Bitansky, Prahladh Harsha, Yuval Ishai, Ron D. Rothblum, and David J. Wu. Dot-Product Proofs and Their Applications. In *FOCS 2024*, pages 806–825, 2024. https://doi.org/10.1109/FOCS61266.2024.00057.

[CFQ19] Matteo Campanelli, Dario Fiore, and Anaïs Querol. LegoSNARK: Modular Design and Composition of Succinct Zero-Knowledge Proofs. In *ACM CCS 2019*, pages 2075–2092, 2019. https://doi.org/10.1145/3319535.3339820.

[DFGK14] George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. Square Span Programs with Applications to Succinct NIZK Arguments. In *ASIACRYPT 2014*, LNCS 8873, pages 532–550, 2014. https://eprint.iacr.org/2014/718.

[DMS26] Michel Dellepere, Pratyush Mishra, and Alireza Shirzad. GARUDA and PARI: Faster and Smaller SNARKs via Equifficient Polynomial Commitments. In *USENIX Security 2026*, 2026. Cryptology ePrint Archive, Paper 2024/1245. https://eprint.iacr.org/2024/1245.

[FLZ13] Prastudy Fauzi, Helger Lipmaa, and Bingsheng Zhang. Efficient Modular NIZK Arguments from Shift and Product. In *CANS 2013*, LNCS 8257, pages 92–121, 2013. Cryptology ePrint Archive, Paper 2012/548. https://eprint.iacr.org/2012/548.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic Span Programs and Succinct NIZKs without PCPs. In *EUROCRYPT 2013*, LNCS 7881, pages 626–645, 2013. https://eprint.iacr.org/2012/215.

[Gro06] Jens Groth. Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures. In *ASIACRYPT 2006*, LNCS 4284, pages 444–459, 2006. https://doi.org/10.1007/11935230_29.

[Gro10] Jens Groth. Short Pairing-Based Non-Interactive Zero-Knowledge Arguments. In *ASIACRYPT 2010*, LNCS 6477, pages 321–340, 2010. https://eprint.iacr.org/2009/390.

[Gro16] Jens Groth. On the Size of Pairing-based Non-interactive Arguments. In *EUROCRYPT 2016, Part II*, LNCS 9666, pages 305–326, 2016. Cryptology ePrint Archive, Paper 2016/260. https://eprint.iacr.org/2016/260.

[GM17] Jens Groth and Mary Maller. Snarky Signatures: Minimal Signatures of Knowledge from Simulation-Extractable SNARKs. In *CRYPTO 2017, Part II*, LNCS 10402, pages 581–612, 2017. https://eprint.iacr.org/2017/540.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect Non-interactive Zero Knowledge for NP. In *EUROCRYPT 2006*, LNCS 4004, pages 339–358, 2006. https://eprint.iacr.org/2005/290.

[KW93] Mauricio Karchmer and Avi Wigderson. On Span Programs. In *8th Annual Structure in Complexity Theory Conference*, pages 102–111, 1993. https://doi.org/10.1109/SCT.1993.336536.

[KZG10] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-Size Commitments to Polynomials and Their Applications. In *ASIACRYPT 2010*, LNCS 6477, pages 177–194, 2010. https://doi.org/10.1007/978-3-642-17373-8_11.

[KPS26] Shubham Khurana, Sahadeo Padhye, and Rajeev Anand Sahu. Zero-Knowledge Extension of PARI. *IACR Communications in Cryptology*, 3(2), 2026. https://doi.org/10.62056/ahjb0l2hd.

[Lip12] Helger Lipmaa. Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments. In *TCC 2012*, LNCS 7194, pages 169–189, 2012. https://doi.org/10.1007/978-3-642-28914-9_10.

[Lip13] Helger Lipmaa. Succinct Non-Interactive Zero Knowledge Arguments from Span Programs and Linear Error-Correcting Codes. In *ASIACRYPT 2013, Part I*, LNCS 8269, pages 41–60, 2013. https://doi.org/10.1007/978-3-642-42033-7_3.

[Lip22] Helger Lipmaa. A Unified Framework for Non-universal SNARKs. In *PKC 2022, Part I*, LNCS 13177, pages 553–583, 2022. https://doi.org/10.1007/978-3-030-97121-2_20.

[Lip24] Helger Lipmaa. Polymath: Groth16 Is Not the Limit. In *CRYPTO 2024, Part X*, LNCS 14929, pages 170–206, 2024. https://eprint.iacr.org/2024/916.

[Nef01] C. Andrew Neff. A Verifiable Secret Shuffle and Its Application to E-Voting. In *ACM CCS 2001*, pages 116–125, 2001. https://doi.org/10.1145/501983.502000.

[SW14] Amit Sahai and Brent Waters. How to Use Indistinguishability Obfuscation: Deniable Encryption, and More. In *STOC 2014*, pages 475–484, 2014. https://eprint.iacr.org/2013/454.

[Val76] Leslie G. Valiant. Universal Circuits (Preliminary Report). In *STOC 1976*, pages 196–203, 1976. https://doi.org/10.1145/800113.803649.